DEVELOPMENT OF A FIELDBUS COMPATIBLE REMOTE DATA ACQUISITION AND CONTROLLER SYSTEM

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

Master of Technology

by

APURVA GAIWAK

to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

February, 1994

CERTIFICATE

It is certified that the work contained in the thesis entitled "Development of a Fieldbus compatible Remote Data Acquisition and Controller System", by "Apurva Gaiwak" has been carried out under our supervision, and that this work has not been submitted elsewhere for a degree.

Dr. K. R. Srivathsan
Professor
Dept. : EE
I.I.T. Kanpur

Dr. A. Joshi
Professor
Dept. : EE
I.I.T. Kanpur

February, 1994

ABSTRACT

With increasing sophistication in supervisory control and data acquisition systems (SCADA), used to monitor and control process plants, cost effective and versatile industrial networks have become a necessary part of industry. Until now networks supplied by leading process control equipment manufacturers have provided proprietary solution for industrial automation. Recently leading standards bodies such as Instrument Society of America, International Electrotechnical Commission and International Standards Organization have taken efforts to provide a set of open standards called FieldBus for the evolution and growth of future industrial networks.

In this thesis, a brief overview of some well known proprietary industrial networks of leading industrial automation equipment vendors is presented. This is followed by a summary of the current draft proposal of Fieldbus standards. An architecture for a microcontroller based Remote Transmission Unit, which can be used for a networkable remote SCADA field unit is suggested. Based on this a remote transmission unit, using 68HC11E9 microcontroller, has been designed and tested.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AE	Application Entity
AE-I	Application Entity Invocation
AIDU	Applicaion Interface Data Unit
AL	Application Layer
ALMIB	Application Layer Management Informat Base
ALS	Application Layer Service
ALSI	Application Layer Service Interface
ALU	Application Layer User
AO	Application Object
AP	Application Process
BPDU	Bridge-protocol-data-unit
DL	Data Link Layer
DLC	Data Link Connection
DLCEP	Data Link Connection End Point
DLE	Data Link Entity
DLL	Data Link Layer
DLP	Data Link Protocol
DLPCI	Data Link Protocol-control-information
DLPDU	DL-protocol-data-unit
DLS	Data Link Service
DLSAP	Data Link-service-access-point
DLSDU	DL-service-data-unit
DMAP	Domain Management Application Process
FAL	Fieldbus Application Layer
FC	Frame Control
FCS	Frame Check Sequence
FDC	Fractional Duty Cycle
FD	Fieldbus Device
FDLL	Fieldbus Data Link Layer
FIFO	First-in First-out
F_MIB	Fieldbus Management Information Base

IEC	International Electrical Committee
I/O	Input/Output
LAS	Link Active Scheduler
LLC	Logical Link Control
LM	Link Master
MAC	Medium Access Control
OSI	Open Systems Interconnection
PAI	Protocol Addressing Information
PCI	Protocol Control Information
pDLSDU	partial DL-service-data-unit
PDU	Protocol Data Unit
PhE	Physical Entity
PhICI	Physical-interface-control-information
PhID	Physical-interface-data
PhIDU	Ph-interface-data-unit
PhL	Physical Layer
PhS	Physical Service
PhSAP	Physical-service-access-point
QoS	Quality of Service
SLAE	Systems-load-application-entity
SMAE	Systems-management-application-entity
SPDU	Support-protocol-data-unit

INTRODUCTION

1.1 Introduction

There are three main flows which influence the industrial processes :

1. Material flow
2. Energy flow and
3. Information flow

The basic objective of plant automation is to regulate the material and energy flow through the information flow, in a desired optimal manner.

Many factors such as VLSI technology, programmable controllers, modern control theory, artificial intelligence, and standardization of data communications links and networks have contributed to the development of modern automation industry.

In this thesis, an overview of the standardization and development of data communication networks and links for industrial applications has been included.

The original concept of centralized point-to-point connection of sensors and actuators to the central room has proven to be highly rigid and very expensive one. Thus, it has had to be improved according to the new design concepts of automation systems. Modern designs favor the decentralized structures that contribute to the minimization of wiring and cable laying

expenses, and at the same time improves the diagnostics,the reliability, and the availability of the automation system itself.

In modern, computer based distributed automation systems, data communication between individual system parts play an important role. Here, process data has to be exchanged between the intelligent monitoring and control stations. Bus or a data highway based organization of distributed control and data acquisition system enable modular and structured development of different physical and logical components. This approach has led to increasing use of local area networks in industrial automation process.

When applied to the industrial automation systems, LANs should meet the following minimum requirements:

* High network reliability and availability
* Low error rate of data transfer
* Self diagnostic features
* Connections to the individual equipments, to other networks and other bus systems
* Maintainability
* Low installation and reconfiguration costs

Apart from these requirements an industrial network should be able to support "Real Time Applications" and should meet the noise and safety requirements applicable in the industry.

A variety of networks are used in industrial applications, some proprietary and some based on standards[1].

Industrial networks and related standards are briefly discussed in chapter 2.

Emergence of OSI[2] standards and standardization of LANs, with requirement of cost effective network and data communication links for real time distributed data acquisition and control, have led to the development of FIELDBUS standards [3] by ISA, IEC and ISO[4]. These standards are expected to have many features in common with different proprietary fieldbuses like, BITBUS, FIP, MIL-STD-1553, and PROFIBUS [3].

Fieldbus; It is an all-digital, two-way, multi-drop communications system used for communication among field instruments and control room systems. Field instruments can include transmitter, control valves, on/off valves, motors, pumps, scales, bar-code readers, multiplexers, multi-loop controllers, PLCs, hand-held communicators, and so on. While referring to fieldbus one must keep in mind that it is not intended to replace plant wide or control system LANs. Instead, fieldbus is designed to be used as the lowest-level communications system in a plant. In contrast to plant-wide or control system LANs, fieldbus has been designed to meet the requirements of the plant floor, such as noise immunity, intrinsic safety, and power and communications media to support two wire devices. One of the key attributes of fieldbus is that it supports two-way communication of multiple variables in a field device. This a major advantage over traditional communications based on analog or discrete techniques, such as 4-20 mA analog current loops. Traditional approaches allow one variable to be transmitted in one direction only, while in the

fieldbus bidirectional transmission is possible.

As shown in the architecture of the fieldbus, [Fig 1.1], three types of wiring topologies are possible in the fieldbus[4]'

(a) Point to Point, where each field instrument is dedicated to a low speed fieldbus, operating at 31.25 kbits/s.

(b) Tree or Multi point, where several field instruments can be connected to a single low speed fieldbus, that runs to the control room.

(c) High speed fieldbus with bridges, in this topology a high speed bus (1.0 or 2.5 Mbits/s), running from the control room to the field, connects different field instruments and other low speed buses. This type of topology is called Multi drop configuration. In the this topology Bridges are used to connect a high speed bus with a low speed bus.

In this work a review of the fieldbus standards, and the development of a fieldbus compatible remote data acquisition and controller card are presented.

1.2 Organization Of Thesis

Chapter 2 gives a general overview of LANs in industry. This is followed by a brief survey of commercially available industrial networks.In the last section of this chapter Manufacturing Automation protocol is discussed.

Chapter 3 presents a brief introduction of the fieldbus standards.

In Chapter 4 development of a remote data acquisition

and transmission unit is discussed.

Chapter 5 summarizes the work done in this thesis and suggests on some future work in this area.

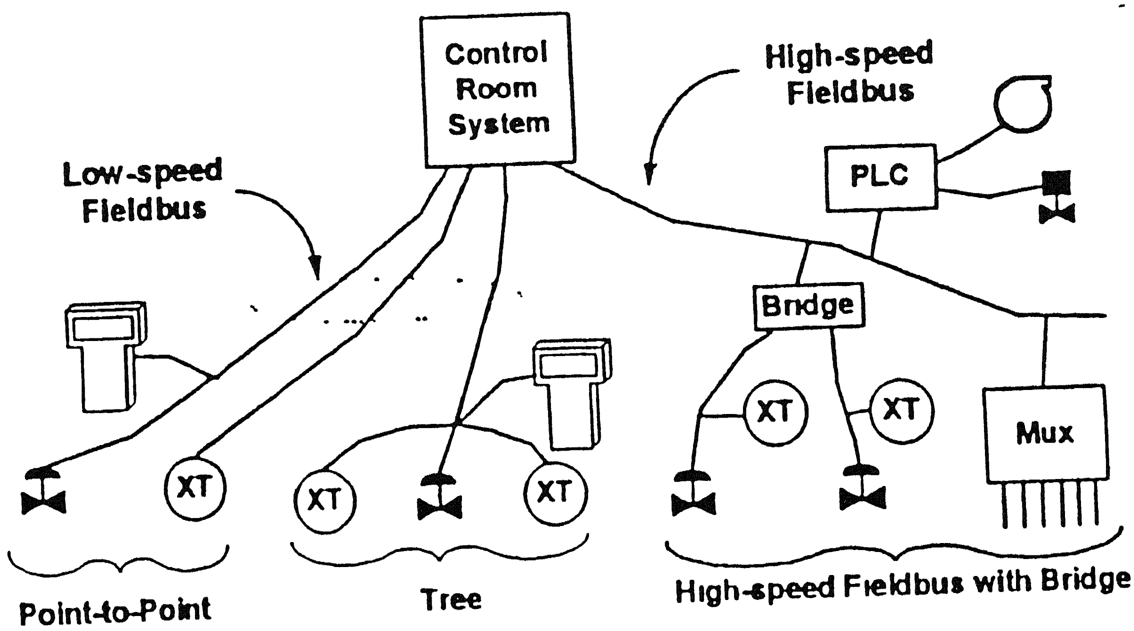


Fig. 1.1 Architecture of Fieldbus Based System

CHAPTER 2

OVERVIEW OF INDUSTRIAL NETWORKS

The term industrial automation refers to the science and technology of process control and includes the control of chemical and petrochemical plants, oil refineries, iron and steel plants, power plants, cement mills, paper mills, pharmaceuticals, food and beverages industries, water and waste water treatment plants, oil and gas fields, etc. In this chapter, a general introduction of LAN in is presented. This is followed by a brief survey of commercially available industrial networks (section 2.2). An introduction to MAP is given in section 2.3.

2.1 An Overview Of local Area Networks(LAN)

Development of LANs during the 70s, primarily to meet some specific requirements, for realizing the multi computer systems(computer networks), wherein a series of computer-based, intelligent terminals spread over a distance of 100 meters to a few kilometers, are interconnected using bit serial communication links.

The International Standards organization(ISO) developed a model, called Open System Interconnection(OSI) model, which abstract the functions in the interconnection of the intelligent communication equipments. The OSI model views the various protocols required to achieve communication between application process in two remote hosts connected over a network as a set of

layers, Fig 2.1. Based on the services provided by each layer and functions implemented in them, they can be divided in two groups, Application oriented layers (upper layers) and Network oriented layers (lower layers).

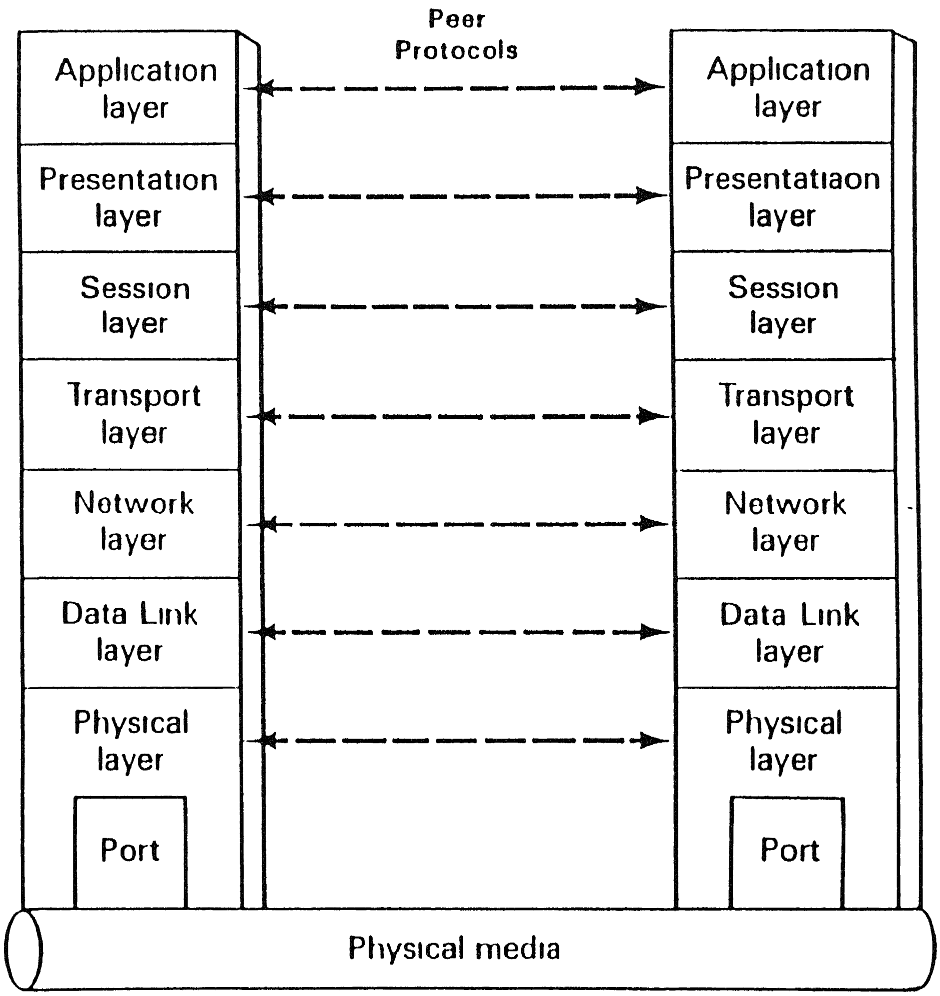


Fig. 2.1 OSI Layers

Most common networks are characterized by a "network" standard (e.g. Ethernet, IEEE 802.5 Token Ring, FDDI, X.25 etc.) Fig 2.2, and an upper layer protocol stack above data link layer (e.g. TCP/IP, DECNET, SNA etc.). Common networks such as TCP/IP over ethernet or SNA over Token Ring support primarily computing and message passing environment.

	IEEE 802	ISO / TC97
LLC	IEEE Standard (September 1983) 802 2	Draft International Standard (October 1984) DIS 8802/2
CSMA / CD	IEEE - Standard (June 1983) 802 3	Draft International Standard (October 1984) DIS 8802/3
Token Bus	IEEE - Standard (September 1983) 802 4	Draft International Standard (October 1984) DIS 8802/4
Token Ring	IEEE Standard (November 1984) 802 5	Draft Proposal (October 1984) DIS 8802/5
Slotted Ring	Nil	Draft Proposal (October 1984) DIS 8802/6

Fig. 2.2 Recommendations of IEEE 802

Industrial networks are used to connect different computers and Supervisory Control and Data Acquisition Systems (SCADA). These industrial networks are required to support the time critical message passing and exceptions reporting (alarms).They are also used for non time critical message passing.

As may be seen from the review of the features of several proprietary industrial networks presented in the following section, many of them use two independent physical networks (often called buses), one to pass the time critical message and alarms and other for non time critical functions.

2.2 A Review of Industrial Networks[4][8]

The following paragraphs give a brief review of industrial networks for DCCS. A list of network names and the manufacturers is given in table 2.1. As shown in the this list many of these networks use bus structure, either twin bus or multi bus, at different data rates. A few of these networks use ring topology also. While coaxial cable is the most commonly used transmission medium, some of these networks use fiber optic link as a local bus. The following discussion gives a brief introduction of different features of the commercially available networks, a comparison of these features is given in table 2.1.

2.2.1 TDC 2000

TDC 2000 system of Honeywell, Fig 2.3, introduced in 1975, has a bus oriented structure, with DATA HIWAY as a central communication path, via this the system operational modules are able to exchange the data. DATA HIWAY of TDC 2000 uses coax cables

as transmission medium. It can have up to three branches of 1.5 km each, and can integrate up to 63 ports. Data communication over the branches is coordinated by the Hiway Traffic Director, which in addition to this serves the I/O devices by polling the bus requests and monitors the bus availability.

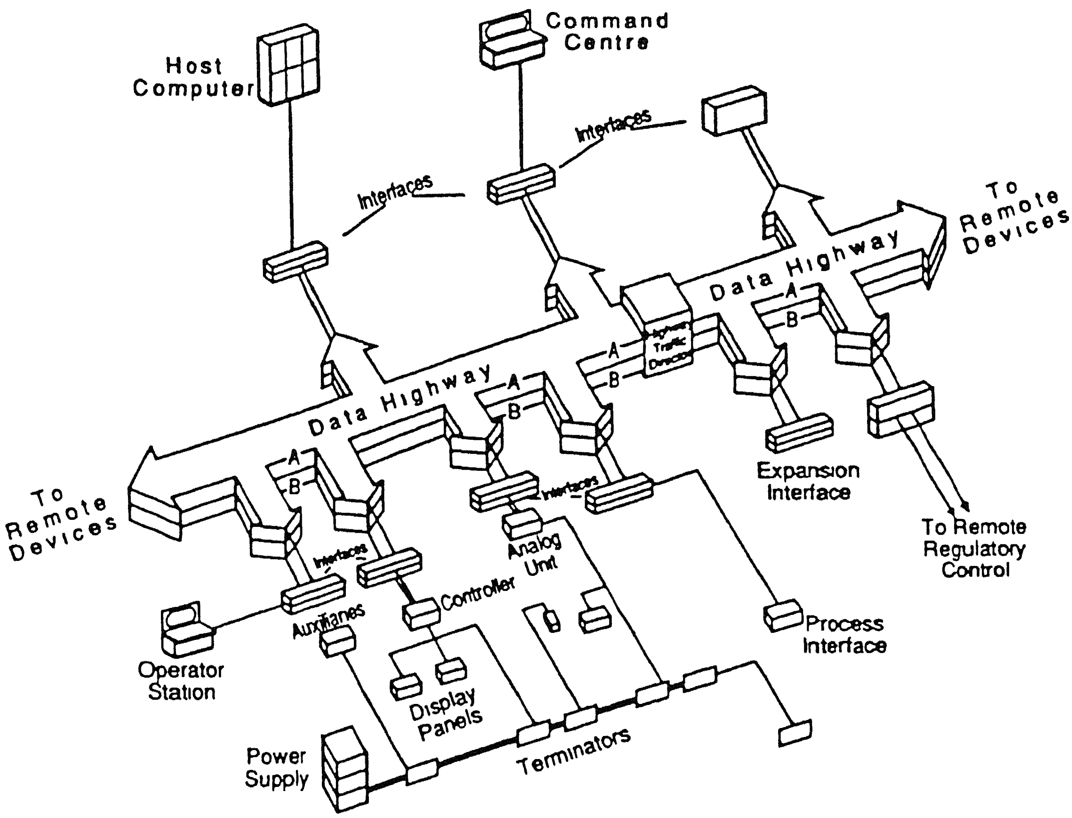


Fig. 2.3 Bus-Oriented Structure of TDC 2000

2.2.2 Spectrum

Foxboro introduced the distributed computer control system SPECTRUM,Fig 2.4 soon after Honeywell had introduced its TDC 2000. It made use of two different industrial system buses to build what is known as FOXNET, A flexible, reliable, and high-speed communication sub-system. Some additional features of FOXNET are:

easy system configuration, re-configuration and extension

maximum transfer rate of 1 Mbps, via a coax cable as transfer medium, at distances up to 4.5 km

integration of up to 100 stations by attachment to the communication sub-system

The two buses of the FOXNET are : Short distance bus and Long Distance Bus, the first one is a communication link within the individual cluster configurations of SPECTRUM, which provides the centralized control of a plant by enabling the direct interfacing of control devices to the central control room.

For realization of a distributed system configuration, a high-speed serial communication link is used to which up to 10 clusters can be attached via the corresponding connectors. Each cluster contains-connected via a LINKPORT - up to 10 local stations at distance of up to 150 m. For data transfer a protocol is used which is similar to the SDLC, it has the frame format with fixed control fields and with variable data field. In the system, a transfer rate of 1 Mbps is achievable with the check of header and longitudinal parity, as well as of cyclic redundancy. Beside this, the functionality of bus lines is monitored and, in case of the line break, the redundant line is automatically switched.

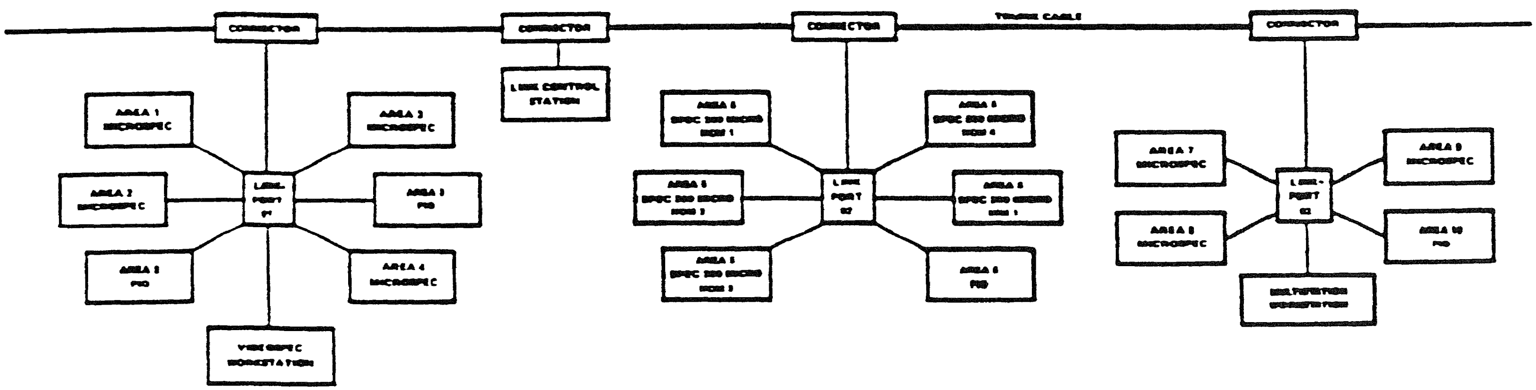


Fig. 2.4 FOXNET-Oriented Structure of SPECTRUM

2.2.3 Provox

In PROVOX system of Fisher Controls also uses two buses, here one bus is used for networking and other for interconnections.

As shown in the Fig 2.5 to remote Bus, or Network Highway, which is driven by a Network Traffic Director (NTD), up to six Network Sub-Systems(NSS) and up to 8 Local Traffic Directors(LTD) can be connected, each of them being able to serve one Local Highway with up to 30 Local Sub-Systems. Both Highway can have a maximal length of 1.5 km and maximal transfer rate of 250 Kbps.

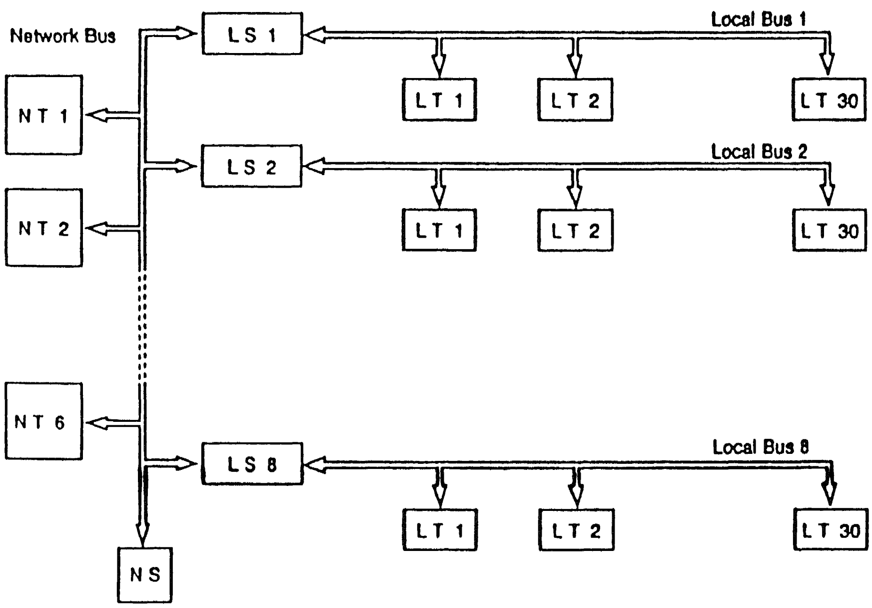


Fig. 2.5 PROVOX Structure

2.2.4 P 4000

In the P 4000 Distributed Control System of Kent, there are two Data Transport Systems(DTS), interconnected by an Inter System Link. On the process side dual serial link is used.

In the system up to 4 independent paths of the DTS can be used for data transfer with transfer rate of up to 1 Mbps. Here also, for reliable data transfer, the redundant system configuration can be used.

2.2.5 Micon MDC 200

In MICON MDC 200-system of VDO,up to four Sub-Communicators and other system modules can be attached, with each up to 8 Local Control Systems.

2.2.6 Teleperm M

A combination of a Remote and a Local bus is also used in the TELEPERM M system of Siemens, in which the Remote Bus interconnects the local buses of the automation stations. The length of the Local Bus can be up to 100 m. Both the local as well as the Remote Bus can be implemented in redundancy if required. For bus to bus interconnection, a bus coupler unit is used.

2.2.7 Procontrol I

Another DCCS, applying the two-bus concept is PROCONTROL I system of BBC. The buses, used here are, The Remote Bus (PROCONTROL I44), by means of which the individual system parts are interconnected, and the Feeder Bus (PROCONTROL I41), by means of which the process data are collected in the field by attaching different units to the Bus.

The Remote Bus of PROCONTROL I has as transfer medium a

screened twin-axial cable and is redundantly implemented. Its maximal length is 3 km and the data transfer rate 1 Mbps. The Feeder Bus, the maximum length of which is 1 km, and transfer rate 500 Kbps, can interconnect up to 50 Stations.

In PROCONTROL I System of BBC an additional bus is used for interconnection of Multi-Purpose Process Stations (PROCONTROL I 14), the so-called Local Bus, with maximal length of 50m, and maximal transfer rate 1 Mbps. In the system, the interconnection between the Feeder Bus and the Local Stations (PROCONTROL I) needs a Branching Device and a Data Link Controller to control the data exchange between the Feeder Bus and the Local Stations, separating in this way the data transmission for further processing within the Stations. Finally, for connections to the Remote Bus, a Remote Controller is applied.

2.2.8 PLS 80 and DCI 4000

In PLS 80-system of Eckhardt or in the new version of DCI 4000 system of Fisher and Porter, for system integration the local buses with optical phase as transmission medium are used.In the PLS 80 system the local buses, maximum transfer rate of which is 500 Kbps, are interconnected via a Remote Bus Coupler (RBC), through which a distance of up to 4 km can be spanned. In the system, up to 40 Communication Units can be connected to each Local Bus.Both local buses, including the coupling between them, are implemented in a redundancy, with the HDLC data transfer protocol for the bus arbitration.

2.2.9 DCI 5000

DCI 5000 Fig 2.6 system of Fischer and Porter, is designed to meet both the requirements of DCCSas well as that of a Data Management, the network with the maximal length of 50km is used as the basic communication link, to which the Direct Control Units are connected. for longer distances, a Bus Repeater or a Remote Bus Repeater, operating in optical fiber as transmission medium, is used for point-to-point interconnection of two network segments.

For interconnection of host computers to the automation system a Gateway (GTW) is available.

Network, used within the DCI 5000, has a coaxial cable as transfer medium for which the CSMA/CD procedure is applied. The cable is closed at both ends by proper terminating resistors and enables a transfer rate of 10 Mbps.

2.2.10 Logistat CP 80

The LOGISTAT CP 80 system of AEG-TELEFUNKEN belongs to the multi-bus system of DCCS. In the system, different communication links are applied at different hierarchical levels, the most important of them being

K 110, a low-cost coupling system for star connection at lowest hierarchical levels, and for distances of up to 1000m

K 120, a bus system for direct interconnection of sub-systems (up to 5km)

K 200, a high-speed, short-distance coupling link

K 300, a long-distance, low-speed coupling system

K 400 a long-distance, very high-speed coupling system with

the HDLC as data transfer protocol.

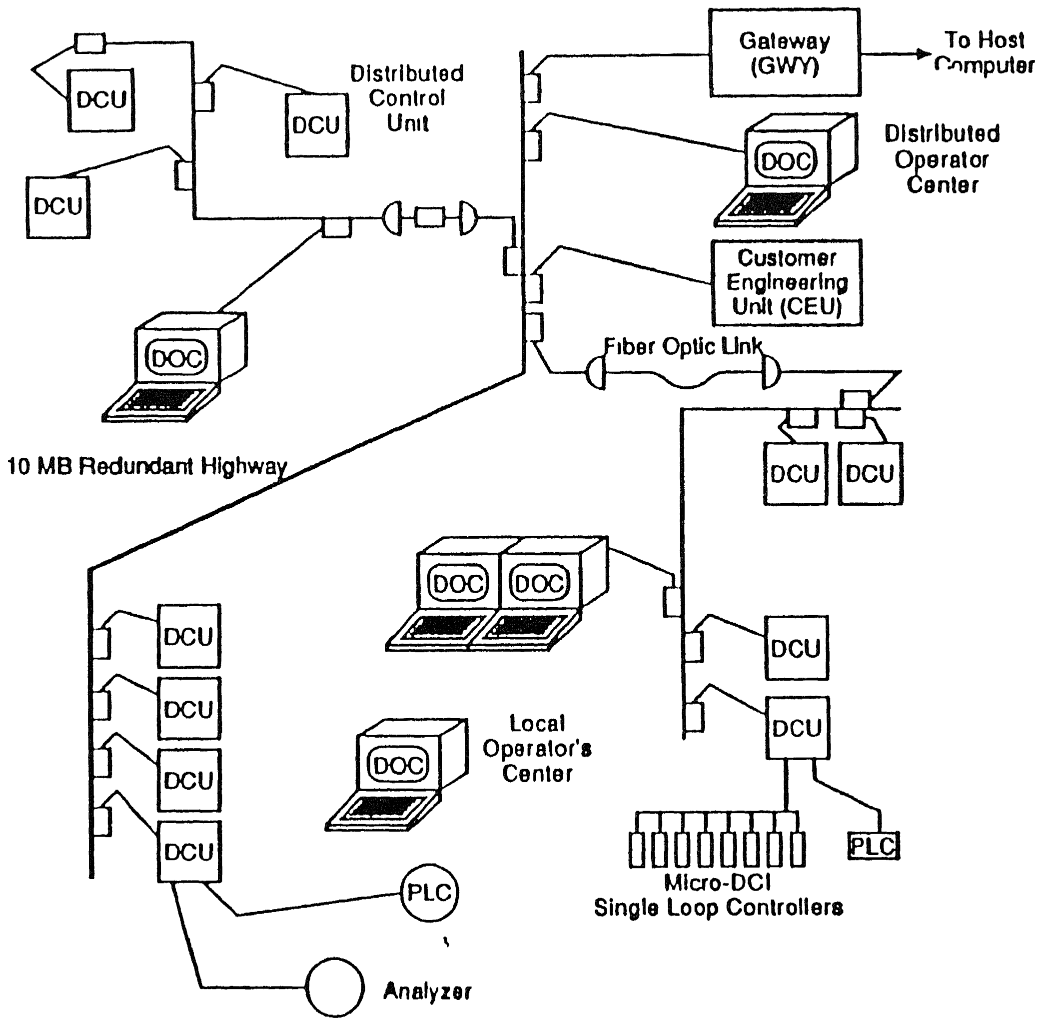


Fig. 2.6 Data Communication System of DCI 5000

2.2.11 TDC 3000

In the TDC 3000 system, Fig 2.7,of Honeywell, in which the many new developmentsin the communication technology are integrated for implementation of efficient and flexible distributed control system. In this system, the Local Communication Network is used as the central communication path, to which the following communication links are connected :

Data Highway (DH), via the Highway Gateway (HG)

Highway Extensions, fiber optic based, via Extension Gateways (XG)

other networks, via Network Gateways (NG)

computers and processors, via relevant Gateways (CC and PG)

The Local Communication Network, whose transfer rate is 5 Mbps, and the maximum length 300 m, is capable to interconnect up to 3000 systems Modules like Universal Stations (US), History Modules (HM) and Application and Coupling Modules (AM and CM). Other Sub-systems, like Operator Stations (OS) as well as various controllers (Basic, Extended, Multi function and Programmable Controllers), are connected to the Data Highway.

2.2.12 MOD 300

A completely LAN oriented DCCS MOD 300 has been developed by Taylor Instruments. The central part of the system is a double-ring, Distributed Communication Network (DCN).

The network is a token passing ring, conformingto the IEEE 802.5 Standard for a maximum transfer rate of 1 Mbps and maximum distance of 27 km. By the use of an Universal Gateway (UG), a VAX

or another computer system can be attached to the ring. Furthermore, by the use of a Network-to-Network-Coupler (NNC) two neighboring Distributed Communication Networks can be interconnected. By the use of a token passing ring of MOD 300 the data are transferred in a highly reliable way by the aid of CRC-algorithm-check, message-check and the code-check.

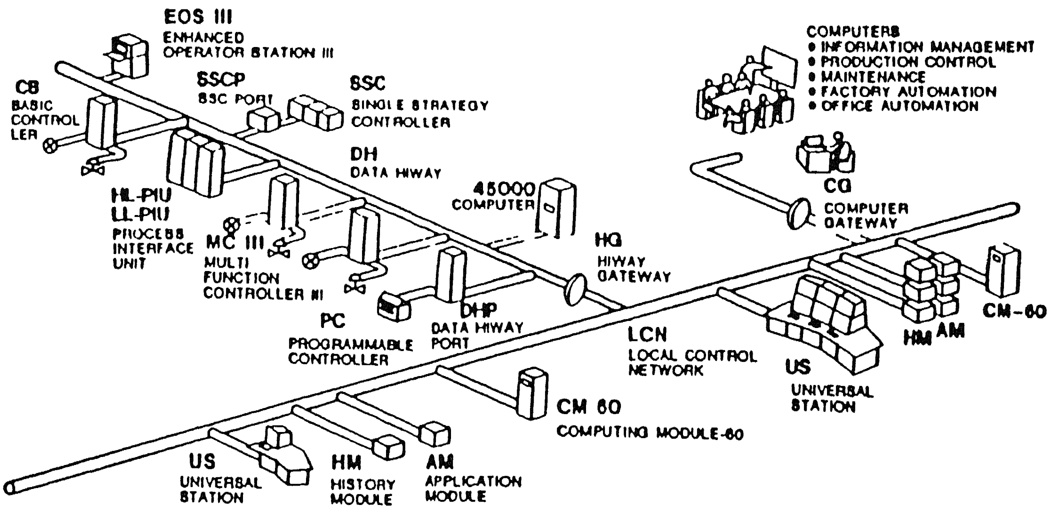


Fig. 2.7 Internal Communication Structure of TDC 3000

2.3.13 MAX 1

MAX 1 of Leeds and Northrup Fig 2.8 is a dual data highway

system, based on a dual optical highway loop, to which the optical/electrical interfaces are connected. From each of the interface a dual redundant electrical highway (wire) can be involved into the multi drop, daisy-chain connection, common to the maximum 15 stations of the system. Each electrical highway spur is also redundant, and can operate as a complete data highway, especially within the station where the noise immunity of the optical highway is not essential.

Bus arbitration, i.e., the access to the high-way by a station, is based on the token passing principle. The data is transferred using the FSK (frequency shift keying) at the rate of 500 Kbaud. Via the Data Highway, whose total length could be up to 22,000 ft, the peer-to-peer communication is possible between the station, the peer being the Controllers, Operator Stations (e.g. CRT Display Terminals), Host Computers, etc. The complete highway system is a highly reliable, completely redundant system in which both optical cables are active at all times and transmit the data in opposite directions around the loop. Thus, the failure of a single communication line or of an optical/electrical interface will not interrupt the transmission. Even if both lines fail in the same highway segment, the communication within the system will still not be endangered.

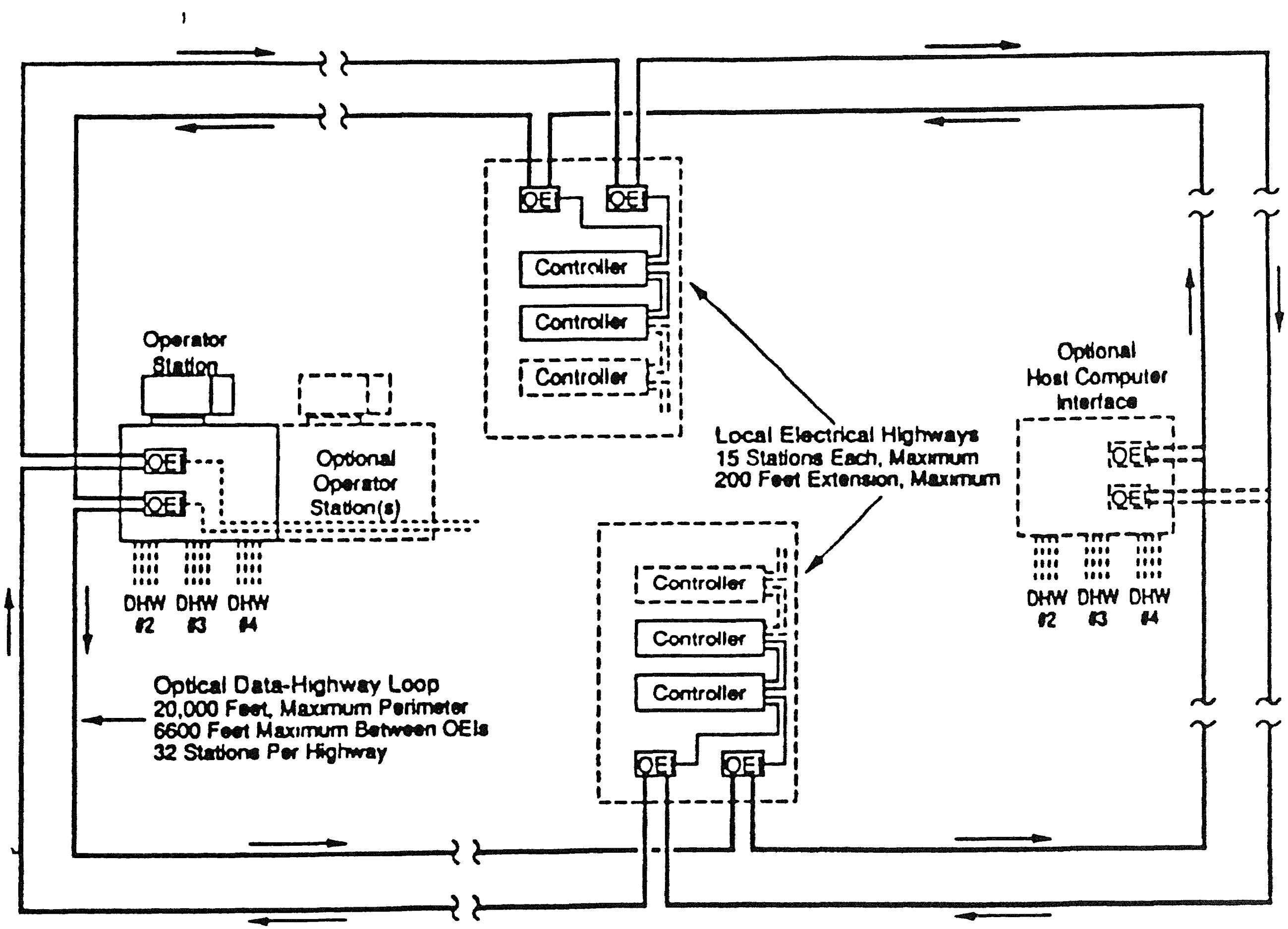


Fig. 2.8 General Structure of MAX 1 System

Table 2.1 Comparison of Different Industrial Ntworks

NAME	PRODUCER	TOPOLOGY	DATA RATE	TRANS MEDIUM	LENGTH
TDC 2000	HONEYWELL	BUS	–	COAXIAL CABLE	1.5KM
SPECTRUM	FOXBORO	TWIN BUS	1MBPS	COAXIAL CABLE	4.5KM
PROVOX	FISHER CONTROLS	TWIN BUS	250 KBPS	COAXIAL CABLE	1.5KM
P4000	KENT	TWIN BUS	1MBPS	COAXIAL CABLE	–
MICON MDC200	VDO	–	–	–	–
TELEPERM M	SIEMANS	TWIN BUS		COAXIAL CABLE	4KM
PROC-CONTROL I	BBC	TWIN BUS	1MBPS	TWIN AXIAL CABLE	3KM
PLS-80	ECKHART	TWIN BUS			
DCI-5000	FISHER AND PORTER	TWIN BUS	10MBPS	COAXIAL CABLE	50KM
LOGISTAT CP-80	AGE TELEFUNKEN	MULTI BUS	–	–	–
TDC-3000	HONEYWELL	MULTI BUS	–	–	–
MOD-300	TAYLOR INSTRUMENTS	RING	1MBPS	–	27KM
MAX-1	LEEDS & NORTHUP	RING	500 KBPS		

2.3 MAP - Manufacturing Automation Protocol[2][4]

As discussed in the previous section, each network provided by different manufacturers uses different architecture and different data rates. A major problem of planning of new automation systems consists in selecting the best, but low price equipment. A way of reducing cost of networking and automation would be to apply a universal, vendor independent communication interface within the automation system. This was tried by General Motors in 1980, by installing the MAP task force. The objective of MAP is to define a local network and associated communications architecture for terminals, computing resources, programmable devices, and robots within a plant or a complex. It sets standards for procurement and provides a specification for use by vendors who want to build networking products for factory use that are acceptable to the MAP participants. The strategy has three parts:

(1) For case in which international standards exist, select those alternatives and options that best suits to the needs of MAP participants.

(2) For standards currently under development, participate in the standards-making process to represent the requirement of the MAP participants.

(3) In those cases where no appropriate standards exist, recommend interim standards until the international standards are developed.

Keeping in tune with the above strategies, MAP also

specifies the seven layer ISO-OSI model. MAP specifies those standards and options within standards appropriate for factory environment. Use of MAP has benefited to the network equipment manufacturer by providing a large market and at the same time the user by the reduction in the cost of networking. As an example, approximately 20,000 programmable controllers and 2000 robots were installed at General Motors,and integrated via the appropriate communication links into a MAP based system. This led to the reduction in the cost to the tune of 50% of the total installation cost [3]. Table 2.2 gives standards used by MAP for different layers.

Table 2.2 MAP Standards

Application	ISO DIS 8571 FTAM (File Transfer Access and Management) Subset ISO DIS 8649/1-2 and 8650/1-2 ACSE (Application Control Service Elements) ISO DIS 9495 Directory Services ISO DIS 9596 Network Management ISO DIS 9506/EIA RS 511 MMS (Manufacturing Message Service)	CCITT X 400 MHS (Message Handling System) ISO DIS 9041 Virtual Terminal ISO DIS 8613 Document Interchange ISO DIS 8632 Graphic Interchange	
Presentation	ISO IS 8822/8823 Presentation Kernel ISO IS 8824 ASN 1 (Abstract Syntax Notation One)		
Session	ISO IS 8326/8327 Session Kernel, Full Duplex		
Transport	ISO IS 8072/8073 Transport Class 4		
Network	ISO DIS 8348/8473 Connectionless Internet	ES-IS ISO DIS	PLP X 25
Data Link	ISO IS 8802/2(IEEE 802 2) LLC1		HDLC LAP B (X.25)
Physical	ISO IS 8802/4 Token Bus (5 MB Carrier Band) (10 MB Broad Band)	ISO IS 8802/3 CSMA/CD (10 MB) ISO IS 8802/5 Token Ring (4 MB)	X.21 / X.21bis
ISO - OSI	MAP 3 0	TOP 3 0	

CHAPTER 3

FIELDBUS STANDARDS

Efforts to standardize fieldbus have been underway since 1985 by international standards organizations such as the International Electrotechnical Commission, the Instrument Society of America, and other working groups in many countries. The fieldbus standards being developed by these bodies is based around the ISO-OSI model. The fieldbus model is a "mini-stack", as shown in Fig. 3.1, suitable for real time applications. As shown in the figure the fieldbus architecture is base on three layer stack,

Application Layer

Data Link Layer

Physical Layer

In this chapter a brief introduction of different fieldbus layers is presented.

3.1 Overview of The Fieldbus

The Fieldbus is intended to be used in factories and process plants to interconnect primary automation devices and to connect these devices with the control and monitoring equipments located in control rooms.

Primary automation devices are associated with lowest levels of the industrial automation hierarchy and perform a limited set of functions within a definite time window. Some of these

functions include diagnostics, data validation, and handling of multiple inputs and outputs.

These primary automation devices, also termed field devices are located close to the process fluids, the fabricated part, the machine, the operator and the environment. This use of the fieldbus position it at the lowest levels of the computer integrated manufacturing architecture.

Some of the expected benefits in using fieldbus are reduction in wiring, increase in amount of data exchange, wider distribution of control between the primary automation devices and the control room equipments, and the satisfaction of time critical constraints.

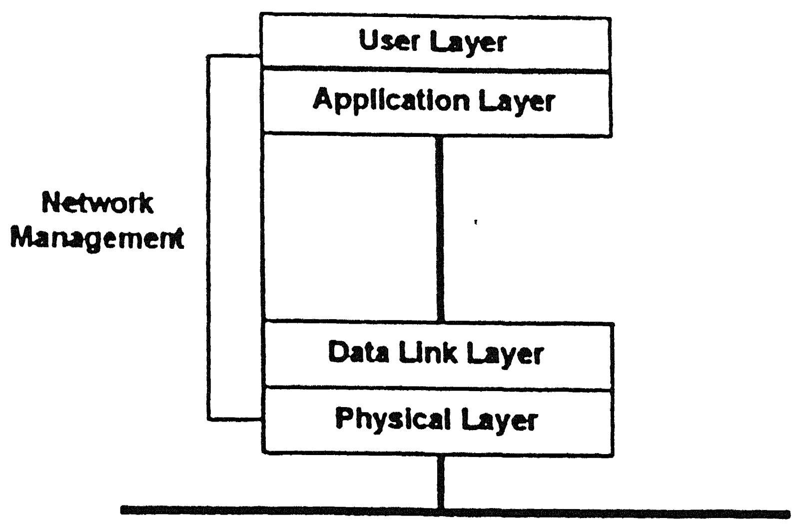


Fig. 3.1 Fieldbus Layers

3.2 Fieldbus Physical Layer [4]

The physical layer of the field bus provides for transparent transmission of data units between Data Link layer entities across physical connections.This layer receives data units from the Data Link Layer, adds preamble and transmits the resulting physical signals to the transmission medium. Signals are then received at one or more nodes, decoded and stripped of the preamble and delimiters, before being passed to the Data Link Layer of the receiving device. A few important characteristics of the physical media are,

Wire media

Digital data transmission

Self clocking

Half duplex communications

Manchester coding

The other variations of the wire media are

Voltage mode (parallel coupling), 31.25 kbits/s

Voltage mode (parallel coupling), 1.0 Mbits/s

Current mode (serial coupling), 1.0 Mbits/s

Voltage mode (parallel coupling), 2.5 Mbits/s

In the fieldbus communication element is considered to be implemented in two parts, the DTE and DCE. The DTE includes only one part of the Physical layer, the DCE independent sub layer(DIS). The DIS transfers the Interface Data Units across a Data Link Layer-Physical layer interface that is not exposed to the user. The DIS then passes the interface data as a serial streams of binary Physical layer service data units across the

DCE-DTE interface, which may optionally be exposed to the user, to a Medium Dependent Sublayer(MDS). The Fieldbus Physical layer conforms to layer 1 of the OSI model with the exception that the frame delimiters are in the Physical Layer.

Station Management-Physical Layer Interface:

This interface provides services to the Physical Layer that are required for initialization and selection of options.The Physical layer should allow the future variations. This calls for a general form of Station Management-Physical Layer interface, which can support different variation of the physical media, such as radio, fiber optics and different modulation techniques.

DCE Independent Sub layer (DIS)

The physical entity is partitioned in to DCE component and DTE component. The DTE component interfaces with the DLL entity and forms DCE independent sub layer (DIS). It exchanges IDUs across DL-Ph interface. This sub layer is independent of all the Physical layer variations, including encoding, modulation, speed, signal, medium etc. All these variations are grouped under DCE. The DIS sequences the transmission of PhID as a sequence of PhSDUs. Similarly, the DIS form the PhID to be reported to the DLL from the sequence of the received PhSDUs. The PhID is converted to a sequence of PhSDUs for serial transmission in octets up to a maximum of 300 octets. A PhSDU representing more significant PhID, transmitted first.

DTE - DCE interface

The Physical Layer entity is partitioned into a DTE component

containing the MIS and higher layers and DCE component containing the MDS and lower sub-layers. The DTE-DCE interface connects these two physical components. This interface may or may not be exposed. The DTE-DCE interface is a functional and electrical, but not mechanical, interface that supports a set of services. Each of these services is implemented by a sequence of defined signaling interface.

<u>Medium Dependent Sublayer (MDS) : Wire media</u>

The MDS is part of the DCE. It exchanges serial PhSDU sequence across the DTE-DCE interface and communicates encoded bits across MDS-MAU interface. The MDS functions are logical encoding and decoding for transmission and reception, respectively, and the addition/removal of preamble and delimiters together with timing and synchronization functions.

<u>MDS - MAU interface : Wire media</u>

The Medium Attachment Unit (MAU) is an optionally separate part of communication element which connects to the medium directly or through passive components. For electrical signaling variations the MAU is the transceiver, which provides level shifting and wave shaping for transmitted and received signals. The MDS-MAU interface links the MAU to the MDS.

<u>Medium Attachment Unit (MAU)</u> : 31.25 kbits/s, voltage mode, wire medium.

The 31.25 kbits/s voltage-mode MAU simultaneously provide access to a communication network and to an optional power distribution network. Devices attached to the network communicate via the medium and may or may not be powered from it. If it isbus

powered, power is distributed as direct voltage and current, and communication signals are superimposed over the DC power. In intrinsically safe applications, available power may limit the number of devices. The network medium consists of twisted pair cable. Independent of topology, all attached devices, other than possibly the transmitting device, are high impedance to prevent significant network loading. Trapezoidal waveforms are used to reduce electromagnetic emissions. Both the bus and tree topologies are supported. In either topology the network contains one trunk cable, terminated at both the ends. In the bus topology spurs are distributed along the length of the trunk, while in the tree topology, spurs are concentrated at one end of the trunk. A spur may connect more than one device to the network, the number devices depending on the length of the spur. At the power frequency the devices appear to the network as current sinks, with minimum current drawn from the medium.

<u>Medium Attachment Unit (MAU)</u> : 1.0 Mbits/s, voltage-mode, wire medium

The 1.0 Mbits/s voltage-mode MAU requirements are not intended to facilitate the options of power distribution via the signal conductors and suitability for intrinsic safety certification. If bus powered, power is distributed as direct voltage and current, and communications signals are superimposed on the DC power.

The network medium consists of shielded twisted pair cable. Like the previous mode all the devices accept the transmitting devices are high impedance to prevent loading. This supports a

linear bus topology. <u>Medium Attachment Unit (MAU)</u> : Current mode, wire medium

The 1.0 Mbits/s current mode MAU simultaneously provides access to a communication network and to a power distribution network. Devices attached to the network communicate via the medium and may or may not be powered from it. Power is distributed as a constant AC current. The communication signals are superimposed on the AC power. The network medium consists of shielded twisted pair cable. Trapezoidal waveforms are used to reduce the electromagnetic emission and signal distortion. The devices are connected in series on the bus, where in the voltage-mode variants the devices are in parallel. The no of devices connected are limited by the available power.

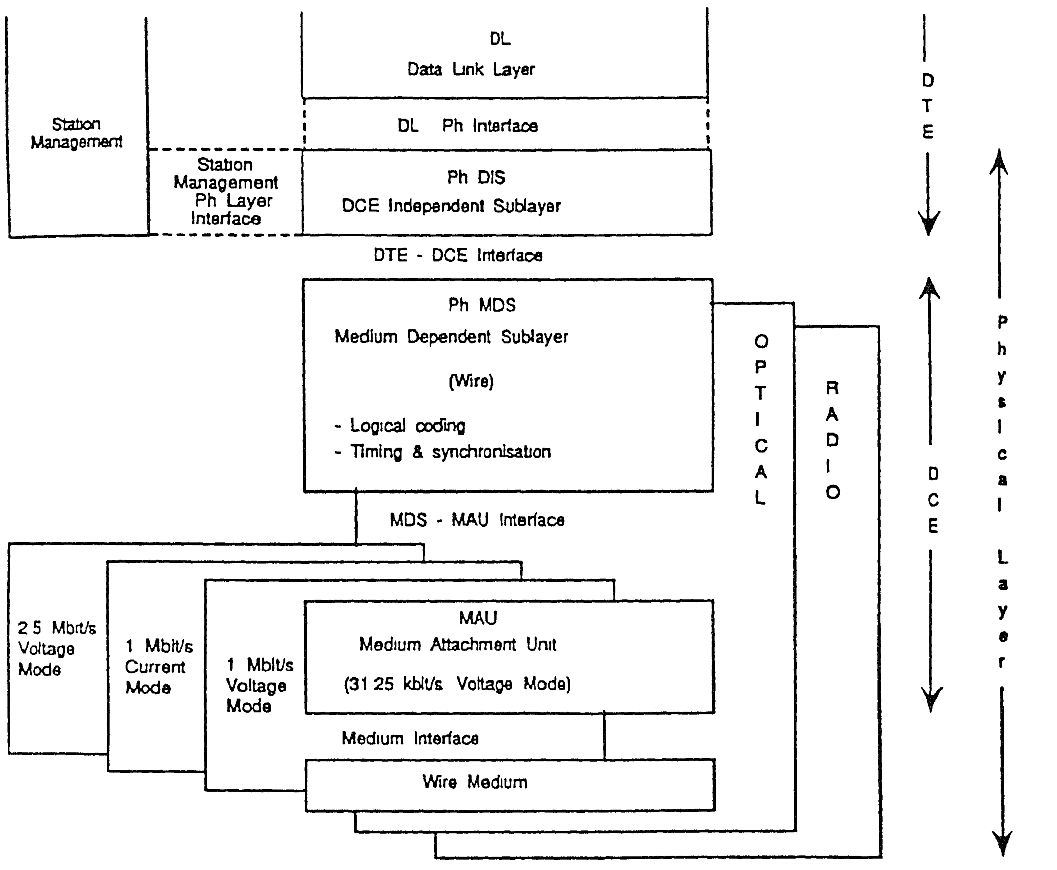


Fig. 3.2 General Model of Fieldbus Physical Layer

3.3 Fieldbus Data Link Layer [5,6]

The data link layer is presented in the following sections under two different headings :

1)..An overview of the Data Link Services and

2)..An overview of the Data Link Protocol

Overview of Data Link Services

The DLS of fieldbus provides for the transparent and reliable transfer of data between DLS-users. It makes invisible to these DLS-users the way in which supporting communications resources are utilized to achieve this transfer.

In particular, the DLS provides for the following:

a) Independence from the underlying Physical Layer. The DLS relieves DLS-users from all concerns regarding which configuration is available(e.g. direct connection, or indirect through one or more bridges) and which physical facilities are used(e.g. which path of a set of diverse physical paths).

b) Transparency of transferred information. The DLS provides for the transparent transfer of DLS-user-data. It does not restrict the content, format or coding of the information, nor does it ever need to interpret the structure or meaning of that information. It may, however, restrict the amount of information which can be transferred as an indivisible unit.

A DLS-user may segment arbitrary-length data into limited-length DLSDUs before making DL-service requests, and may reassemble received DLSDUs into those larger data units.

c) Reliable transfer of data. The DLS relieves the DLS-user from

all concerns regarding insertion or corruption of data, or, if requested, loss, duplication or misordering of data, which may occur. In cases where protection against misordering of data is not employed, misordering can occur.

It is noted that detection of duplicate, lost or misordered DLSDUs may be performed by DLS-users.

d) Quality of Service(QoS) selection. The DLS makes available to DLS-users a means to request and to agree upon a quality of service for the transfer of data. QoS is specified by means of QoS parameters representing characteristics such as mode of operation, transit delay, accuracy and reliability.

e) Addressing. The DLS allows the DLS-user to identify itself and to specify the DLSAPs between which a DLC is to be established. DL-addresses have only regional significance within a specific DL-subsystem over a set of bridged DL-segments. It is noted that the DLS is required to differentiate between the individual systems that are physically or logically connected to a multi-point data link and to differentiate between connections. For commonality with other service definitions, this mechanism is referred to as addressing and the objects used to differentiate between systems are referred to as addresses. In a formal sense, this is an extension of the use of addresses beyond that specified in ISO 7498-3.

f) Scheduling. The DLS allows the DLS-user to provide some guidance on the internal scheduling of the DLS-provider. This guidance supports the time-critical aspects of the DLS, by permitting the DLS-user some degree of management over when

opportunities for communication will be granted to various DLEs for various DLSAPs and DLCEPs.

g) Common time sense. The DLS can provide the DLS user with a sense of time which is common to all DLS-users on the (extended) link.

h) Queues and buffers. The DLS can provide the sending or receiving DLS-user with either a FIFO queue or a retentive buffer, where each queue entry or buffer can hold a single DLSDU.

Overview of DL-sub network structuring

A DL-sub network consists of a set of DL-segments (links) interconnected by DL-relay entities(bridges) which provide DL-layer internal coordination and routing services to the set of interconnected DLEs.

A DL-segment consists of a set of DLEs, all of which are connected directly(that is, not through a DL-relay entity) to a shared logical communications channel, known as a link. A link (logical communication channel) consists of one or more physically-independent separately and cooperatively-scheduled communications channels, which are known as paths.

It is noted that a logical link consisting of more than one path is an instance of DL-redundancy. This is distinct from Ph-redundancy, which is necessarily hidden from the DLL and all DLEs due to the principles of layering. In a logical sense, DLEs are connected to links and bridges interconnect links. Yet in a physical sense, DLEs are connected to paths and bridges interconnect these paths. DL-communication-services are

independent of the specific path employed, and the DLS-user has no cognizance of any path multiplicity.

Overview of DL-naming(addressing)

DL-names, known conventionally as DL-address, are identifiers from a defined identifier space -- the DL-address-space -- which serve to name objects within the scope of the data link layer. Examples of such objects are data-link-service-access-points (DLSAPs),data-link-connection-end-points(DLCEPs),and data-link-entities (DLEs).

The DL-address-space from which DL-addresses are drawn may be partitioned into sub-spaces of DL-addresses:

a) to cluster addresses of the same generic function, such as DLSAP-addresses naming specific DLSAPs, DL-addresses naming groups of DLSAPs, DL-addresses designating one or more DLCEPs, and DL-addresses designating DLEs; or

b) to cluster addresses for administrative purposes, such as addresses which are known to be local to, and allocatable by, a single DLE; or known to be local to a single link (DL - segment) but to not have a known specific DLE-locality; or known to have no implicit DL-locality; or

c) to cluster addresses for routing purposes, such as addresses which are known to be local to a single DL-segment or a single DLE.

Functional partitioning of the DL-address-space

The space of DL-addresses may be partitioned functionally as follows:

a) DLE-specific DLSAP-addresses;

b) other DLSAP-addresses;

These addresses can designate at most one DLSAP within the entire set of DL-interconnected DLEs.

c) group DL-addresses designating a group of DLSAPs;

They are sometimes(incorrectly) referred to as group-DLSAP-addresses.

d) DLE-specific DLCEP-addresses;

e) other DLCEP-addresses;

f) specific aspects of a specific DLE;

g) specific aspects of a group of DLEs.

<u>Administrative partitioning of the DL-address-space</u>

The space of DL-addresses may be partitioned administratively as follows:

a) DLE-specific DL-addresses, which are known to refer to objects within the scope of a specific DLE, and which are allocatable by that DLE;

b) DL-segment(link) specific but DLE-independent DL-addresses, which are known to refer to objects within the scope of a specific DL-segment, and which are allocatable locally by the the DL-address-space administrator for that DL-segment.

c) DL-segment-independent DL-addresses, which are known to refer to objects within the DL-connected set of DL-segments, and which are allocatable by the DL-address-space administrator for the connected set of DL-segments.

A DL-address-space administrator can always allocate a set of

addresses to a subordinate administrator for its sub-administration. For example, the DL-administrator for the entire set of DL-connected DL-segments can allocate a contiguous block of unassigned DL-addresses to the DL-administrator for a specific DL-segment, or to the local administrator within a single DLE.

<u>Routing-related partitioning of the DL-address-space</u>

The space of DL-addresses may be partitioned to assist DL-routing activities as follows:

a) DLE-specific DL-addresses;

b) DL-segment(link) specific but DLE-independent DL-addresses;

c) DL-segment-independent DL-addresses.

<u>Types of Data Link Services</u>

There are four types of DLS:

a) a DL(SAP)-address, queue and buffer management service,

b) a connection-mode data transfer service with four classes of service,

c) a connection less-mode data transfer service, and

d) a time and transaction scheduling service with at least three classes of service.

Out of these four services a DLS-user may choose those most appropriate for use. Within the DLS types, the DLS-user is limited to those classes of service supported by the associated DL-protocol implemented.

Quality of service(QoS) attributes common to multiple types of Data Link Service.

A DLS-user may select, directly or indirectly, many aspects of the various data link services. The term "QoS" refers to those aspects which are under control of the DLS-provider.

Some QoS attributes are selected independently at each instance of DL-service; these are known as dynamic QoS attributes. Other QoS attributes are selected once for an entire class of DL-services; these are known as static QoS attributes.

Four QoS attributes of the DL-data transfer services apply conceptually to both connection-mode and connection less operation. Default values for three of these attributes can be set by DL-management.Two of these four attributes are considered dynamic; their DLSAP-address-related values serve merely as defaults for each appropriate DL-service-invocation, and can be overridden on an instance by instance basis.

It is noted that the existence of multiple levels of defaults QoS attribute values and means of setting those default values, can simplify use of the DL-services. One of the other two attributes is considered static for all DL-services; all relevant connection less DL-service-invocations from a DLSAP-address will inherit the static attribute value associated with that DLSAP-address, and all DLCEP-establishment requests and responses will use the static attribute value as their desired QoS attribute value, subject to the rules of DLC negotiation.

The fourth attribute is static for connection less DL-services, but dynamic for all DLCEP-establishment requests and responses;

its DLSAP-address-related value serves merely as a default for each appropriate DL-service-invocation, and can be overridden on an instance by instance basis.

DLL priority (dynamic QoS attribute)

Each DLCEP establishment request and response, and each connection less data transfer request, specifies an associated DLL priority used in scheduling DLL data transfer services. This DLL priority also determines the maximum amount of DLS-user-data which can be conveyed in a single DLPDU, determined by the DL-protocol specification.

The DL-protocol shall support three DLL priority levels, each of which shall be capable of conveying a specified minimum amount of DLS-user data per appropriate DLPDU. The three DLL priorities, with their corresponding minimum conveyable amounts of DLS-user-data(per DLPDU) are(from highest priority to lowest priority):

a) URGENT -- <=64 DLS-user octets per DLPDU;

b) NORMAL -- <=128 DLS-user octets per DLPDU;

c) TIME-AVAILABLE -- <= 256 DLS-user octets per DLPDU.

The first two of these values are considered time critical priority levels; the third (and last) is considered a non-time-critical priority level. DLCEP establishment may negotiate URGENT to NORMAL or TIME-AVAILABLE, or NORMAL to TIME-AVAILABLE.

The default QoS value can be set by DL-management; when not so set its value is TIME-AVAILABLE.

DLL maximum confirm delay(dynamic QoS attribute)

Each DLCEP establishment request, and each response, specifies upper bounds on the maximum time duration permitted for the completion of

a) the DL-connect, DL-RESET and DL-SUBSCRIBER-QUERY primitives, and, separately,

b) the DL-DATA primitive.

Each connection less service request specifies an upper bounds on the maximum time duration permitted for the completion of either

--- a DL-LISTENER-QUERY primitive, or, separately,

--- a DL-UNITDATA primitive.

Each parameter either has the value UNLIMITED or specifies an interval, in units of 1 ms, from 1ms to 1 min. The value UNLIMITED is defined to be greater than all explicit intervals.

The parameters for the DL-DATA and DL-UNIDATA primitives specify intervals less than or equal to that for the DL-CONNECT, DL-RESET, DL-SUBSCRIBER-QUERY, and DL-LISTENER-QUERY primitives. The intervals specified are the maximum permissible delays between the issuing of the specified request primitives and the issuing of the corresponding confirm primitives.

For DLEs which do not support a time resolution of 1 ms, the requested time interval may be rounded up to the next-greatest multiple of the resolution which the DLE does support, or UNLIMITED if the DLE has no sense of time.

The default QoS values can be set by DL-management; when not so set the value for each of these QoS parameters is UNLIMITED.

DLPDU authentication(static QoS attribute)

Each DLCEP establishment negotiation, and each connection less data transfer, uses this attribute to determine a lower bound on the amount of DL-addressing information used in the DLPDUs which provide the associated DLL data transfer services. This has a slight impact on the residual rate of DLPDU misdelivery; more addressing information reduces the potential for misdelivery.

The two levels specified, with their amounts of DL-addressing information, are:

a) ORDINARY -- each DLPDU includes the minimum amount of addressing information necessary;

b) EXTRA -- each DL-address includes the maximal amount of addressing information possible.

The default QoS value can be set by DL-management; when not so set its value is ORDINARY.

DL-scheduling-policy(semi-static QoS attribute)

this attribute is static for connection less services, but it is dynamic for connection-mode services.

For each DLSAP-address, and each DLCEP, the DLS-user can provide the normal(implicitly-scheduled) DLL policy of providing the requested DL-service as soon as possible, and instead defer any inter-DLS-user communication required by a request or response DLS-primitive until that deferral is released by an involved DLS-user. Each such release, by execution of a DL-COMPEL-SERVICE request, specifying the DLSAP-address or DLCEP, permits the

completion of just a single deferred request or response DLS-primitive.

The two choices are:

a) IMPLICIT -- any required communications with peer DLS-users from this DLSAP-address, or from this DLCEP, will occur as soon as possible;

b) EXPLICIT -- any required communications with peer DLS-user(s) from this DLSAP-address, or from this DLCEP, will occur only when the deferral is explicitly released by an involved DLS-user.

The possible use of previously-scheduled communications opportunities make it possible for this deferral and subsequent release to result in earlier communications with the peer DLS-users than that provided by the IMPLICIT alternative. The default QoS value cannot be set by DL-management; its value is always IMPLICIT.

Overview of the DL-protocol

One can represent the DLL with a three level model:

* a low level of path access and scheduling functions which supports

* an intermediate level of bridge operation functions, which in turn supports

* a higher level of connection mode and connection less data transfer, bridge coordination, and DL-service functions. Interoperating with all three levels are DL-management functions, including bridge and redundant path management functions where relevant. One important point should be noted that in the above

discussion the term "levels" is used in place of sub-layer, this in confirmation of ISO 7498 requirement that when multiple sub-layers are defined, all but one of them must be optional.

This three level partitioning closely resembles the partitioning into lower level "MAC", intermediate-level "bridge-operation" and higher level "LLC" functions found in the ISO/IEC LAN standards , with two significant differences:

*In this protocols, low-level functionality is contained entirely within the DLL as specified by ISO 7489. In contrast the "MAC" functionality of the ISO/IEC LAN protocol spans the lower part of the OSI DLL and upper part of the OSI Physical layer.

* This protocol employs a single level of DL-addressing within the DLL. In contrast, the ISO/IEC LAN protocol employs two levels of DL-addressing, one within the "MAC" and "bridge-operation" functionality and a second within the "LLC" functionality.

<u>Path access and scheduling level</u>

(a) The path access and scheduling level forms each DLPDU from DL-protocol control information and DLS-user data, computes and appends an appropriate frame check sequence (FCS), and passes the whole as a sequence of PhIDUs to the PhE for transmission to peer PhEs for reporting to peer DLEs. In some cases it also appends the lower order two or four octets of the current value of the local node timer, during DLPDU formation, immediately preceding the appended FCS.

(b) The path-access and scheduling level receives a sequence of PhIDUs from the PhE, concatenates those PhIDUs into a received

DLPDU, computes a frame check sequence over the entire sequence of received data, and checks the proper residual value. The first octet of the receive sequence is examine to determine the type of DLPDU, and an attempt is made to parse that DLPDU into its DL-protocol control information and DLS-user data components. If the FCS residual was correct and the parse was successful, then the appropriate low-level actions are performed, possibly including reporting the parsed DLPDU to a higher level. In some cases the value of the lo-order two octet of the local node timer, at the time of receipt,is appended to this parsed DLPDU.

(c) The path-accessed-and-scheduling level provides the basic functions of the both responder and initiator. As a responder, it provides the sequencing functions necessary

(1) for receiving a DLPDU, possibly conveying a reply-token; and

(2) in that latter case (of receiving reply token), for sending a DLPDU as an immediate reply to just received DLPDU.

As an initiator it provides the sequencing functions necessary for

(3) receiving the delegated token;

(4) sending one or more DLPDUs, including those requiring an immediate reply;

(5) receiving such an immediate reply, or inferring its absence; and

(6) returning a delegated token.

(d) The path-access-and-scheduling level provides the low level scheduling functionality required for scheduling DLPDU transmission on a specific path, including an interactions with the local links link active scheduler (LAS) to coordinate the

schedule with other DLEs or to request needed path transmission capacity.

The actions of (a) and (b) are augmented within the bridge to permit the retransmission of a received sequence of the data octets, including the received FCS, with possible constrained alterations of the first octet and compensating alterations of the received FCS, as required, prior to transmission.

The actions of (c) and (d) are based in part on three request-management and scheduling queues

i) a DL-address-specific request queue, associated with the sending address, which is used to manage DLS-user requests originated from that DL-address;

ii) a DL-address-specific schedule-service queue, associated with the sending address, which is used to correlate requests for token delegation to that DL-address with committed SPDU or DLS-user request-related transmission; and

iii) the common DLE LAS-request queue, which is used to manage both SPDUs which are awaiting transmission to the local LAS and isolated requests for expedited IMMEDIATE service.

Some of the more complex scheduling functionality of this level requires, and uses, the services of the upper level by acting as a DLE-internal quasi-DLS-user.

<u>Bridge-operation level</u>

The bridge-operation level provides the intermediate-level functionality of

a) logically inter-connecting multiple local links into a single extended-link by physically interconnecting multiple paths;

b) serving as (possibly-distributed) "time-space-time" switch, providing DLPDU store-and-forward functions to permit communication between DLEs in the extended link which could not otherwise communicate. It is to be noted that this includes the coordination with fractional-duty-cycle (FDC) DLEs necessary to permit alternating periods of "FDC-DLE-awake" and "FDC-DLE-asleep" operation;

c) providing a shared sense of DL-time throughout the extended link; and

d) coordinating local-link scheduling among two or more local links to provide any necessary multi-link coordination of scheduling within the extended link.

Much of the bridge-operation level of functionality is active only in "bridge" DLEs.

<u>Service provided by the DLL</u>

The DLL provides connection less data transfer services for limited-size DLSDUs, connected data transfer services for limited-size DLSDUs, an internally-synchronized time service, scheduling services to control the time allocation of the underlying shared PhL service, and a DL(SAP)-address, queue and buffer management service.

Some relevant QoS attributes are as follows :

<u>QoS -DLCEP class</u>

Each DLCEP establishment request specifies the class of the DLCEP. The three choices for DLCEP-class are :

a) PEER - the DLS -user can exchange DLSDUs with one other peer

DLS-user.

b) PUBLISHER - the DLS-user can send DLSDUs to a set of zero or more associated subscribing DLS-users.

c) SUBSCRIBER - the DLS-user can receive, and request, DLSDUs from the associated publishing DLS-user.

<u>Qos - sending DLCEP data delivery features</u>

Both members of a peer DLC, or the publishing DLS-user of a multi-peer DLC, specify the sending data delivery features of the DLCEP. The four choices for sending DLCEP data delivery features, and their effects, are :

a) CLASSICAL - the DLS-user can send data which will be delivered without loss, duplication or misordering; all relevant DLS-users will be notified of any loss of synchronization on the connection.

b) DISORDERED - the DLs-user can send data which will be delivered immediately upon receipt, without duplication but potentially in a different order than that of the sending DLS -user. All relevant DLS-users will be notified of any unrecoverable loss of DLS-user data.

c) ORDERED - the DLS -user can send data which will be delivered immediately upon receipt, without duplication or misordering, but with potential loss of some DLS-user data. Loss of DLS-user data will not be reported. no attempt will be made by the DLL to recover from such loss.

d) UNORDERED - the DLS -user can send data which will be delivered immediately upon receipt. Loss, duplication and misordering of DOS-user data will not be reported. No attempt will

be made by the DLS-provider to recover from such events.

On a peer DLS, the QoS value for the sending DLCEP data delivery features may be chosen independently for each direction of data transfer.

QoS - maximum DLSDU sizes

Each DLCEP establishment request, and each response, specifies an upper bound on the size (in octets)of DLSDUs which will be offered for transmission, and an upper bound on the size of DLSDUs which are acceptable for reception.

For peer connections the negotiated maximum DLSDU size shall be determined independently for each direction of data transfer as the smallest of that offered by the sender, that permitted by the sender's local DL-management, that permitted by the receiver and that permitted by the receiver's local DL-management.

For multi-peer connections, the negotiated maximum DLSDU size shall be the smaller of that offered by the publisher and that permitted by the publisher's local DL-management. For subscribers of multi-peer connections, the connection shall be refused by the DLS-provider if the maximum DLSDU size established by the publisher is larger than the smaller of that permitted by the subscriber and that permitted by the subscriber's local DL-management.

The sender's offered size may be any value between zero and 16 times the maximum number of DLS-user octets per DLPDU.

QoS - DLCEP buffer-and-queue bindings

Each DLCEP establishment request, and each response, can bind one or two local buffers or specified-depth FIFO queues, created

by DL-CREATE buffer and queue management primitives, to the DLCEP. When these bindings are made for a sending DLS-user of a peer or multi-peer DLCE, they determine the maximum transmit window for that direction of DLC data transfer.

a) One buffer or queue can be bound to a Peer or Publisher DLCEP to convey DLSDUs from the DLS-user to the DLS-provider.

b) One buffer or queue can be bound to a Peer or Subscriber DLCEP to convey DLSDUs from the DLS-provider to the DLS-user.

Binding to a buffer

When a sending buffer is bound to a DLC by a DLS-user :

a) A DL-PUT request primitive overwrites the buffer with a DLSDU.

b) A DL-DATA request primitive causes the transmission of the current contents of the buffer. The DL-DATA request primitive does not itself specify a DLSDU.

c) A DL-BUFFER-SENT indication primitive notifies the DLS-user of the specific DLCEP on which the buffer was transmitted, and to which the buffer is bound, that the buffer was just transmitted.

When a receiving buffer is bound to a DLC by a DLS-user :

d) A DL-GET request primitive copies the DLSDU from the buffer.

e) A DL-DATA indication primitive notifies the DLS-user of the overwriting of the buffer by a newly-received DLSDU. The DL-DATA indication primitive does not itself specify a DLSDU.

Binding to a FIFO queue

When a sending FIFO queue of maximum depth K is bound to a DLC by a DLS-user :

a) A DL-PUT request primitive is not permitted.

b) A DL-DATA request primitive attempts to append a DLSDU to the queue, but fails if the queue already contains K DLSDUs. If the append operation was successful, then the DLSDU will be transmitted when permitted, after all preceding DLSDUs in the queue.

When a receiving FIFO quèue of maximum depth K is bound to a DLC by a DLS-user :

c) A DL-GET request primitive attempts to remove a DLSDU from the queue, but fails if the queue is empty.

d) A DL-DATA indication primitive notifies the receiving DLS-user of the result of attempting to append a newly-received DLSDU to the receive queue. The DL-DATA indication primitive does not itself specify a DLSDU.

A FIFO queue can also be bound to wither the sending (DLS-user to DLS-provider) or receiving (DLS-provider to DLS-user) direction of data transfer, at a specified DL(SAP) address and DLL priority.

Multiple concurrent bindings are permitted as an implementation option, but are not required by this protocol.

<u>Default bindings</u>

When these binding options are not specified, the conventional implicitly-queued sending and direct receiving interfaces between DLS-user and DLS-provider are employed, and each associated DL-DATA primitive contains a DLSDU.

a) DL-PUT and DL-GET request primitives are not permitted.

b) A DL-DATA or DL-UNITDATA request primitive by the sending DLS-user attempts to append a DLSDU to the implicit queue, but fails if the queue is full. If the append operation was successful, then the DLSDU will be transmitted at the first opportunity, after all preceding DLSDUs in the queue.

c) A DL-DATA or DL-UNITDATA indication primitive immediately passes a newly-received DLSDU to the receiving DLS-user. No queuing is provided within the DLL.

Service assumed from the Physical Layer

This sub clause defines the assumed Physical Service (PhS) primitives and their constraints on use by the DLE. Proper layering requires that an (N+1)-layer entity not to be concerned with, and that an (N)-service interface not overly constrain, the means by which an (N)-layer provides its (N)-services. Thus the Ph-service interface does not require DLEs to be aware of internal details of the PhE (for example, preamble, postamble and frame delimiter signal patterns, number of bits per baud), and should not prevent the PhE from using appropriate evolving technologies.

Assumed primitives of the PhS

The granularity of transmission in the Fieldbus protocol is one octet. This is the granularity of PhS-user data exchanged at the PhL - DLL interface.

PhS characteristics reporting service

The PhS is assumed to provide the following service primitive to report essential PhS characteristics used in DLL transmission, reception, and scheduling activities :

Ph-CHARACTERISTICS indication (minimum-data-rate,framing overhead)

where

minimum-data-rate – specifies the effective minimum rate of data conveyance in bits/second, including any timing tolerances. A PhE with a nominal data rate of 1 Mbit/s ± 0.01 % would specify a minimum data rate of 0.9999 Mbit/s.

framing-overhead – specifies the maximum number of bit periods, where period = 1 / data rate used in any transmission for PhPDUs which do not directly convey data (for example, PhPDUs conveying preamble, frame delimiters, postamble, inter-frame "silence", and so on). If the framing overhead is F and two DL message lengths are L_1 and L_2, then the time to send one message of length L_1+F+L_2 will be at least as great as the time required to send two immediately consecutive messages of lengths $L_1$and L_2.

If this framing-overhead is less than the DLE's configured minimum-inter-PDU-delay, then the DLE shall report this discrepancy to DL-management and shall not issue Ph-DATA requests while the discrepancy exists. This restriction prohibits DLE transmission while this discrepancy exists. The DLE's local station management may remedy this discrepancy either by reconfiguring the PhE to a greater framing-overhead, or by reconfiguring to a smaller value, or both.

PhS transmission and reception services

The PhS is assumed to provide the following service primitives for transmission and reception :

Ph-DATA request (class, data)

Ph-DATA indication (class, data)

Ph-DATA confirm (status)

where

class - specifies the Ph-interface-control-information (PhICI) component of the Ph-interface-data-unit (PhIDU). For a Ph-DATA request, its possible values are :

START-OF-ACTIVITY - transmission of the PhPDUs which precede Ph-user data should commence;

DATA - the single-octet value of the associated data parameter should be transmitted as part of a continuous correctly-formed transmission; and

END-OF-DATA-AND-ACTIVITY - the PhPDUs which terminate Ph-users data should be transmitted after the last preceding octet of Ph-user data, culminating in the cessation of active transmission.

For a Ph-DATA indication, its possible values are :

START-OF-ACTIVITY - reception of an apparent transmission from one or more PhEs has commenced;

DATA - the associated data parameter was received as part of a continuous correctly-formed reception;

END-OF-DATA - the ongoing continuous correctly-formed reception of Ph-user data has concluded with correct reception of PhPDUs implying END-OF-DATA;

END-OF-ACTIVITY - the ongoing reception (of an apparent

transmission from one or more PhEs) has concluded, with no further evidence of PhE transmission; and

END-OF-DATA-ACTIVITY - simultaneous occurrence of END-OF-DATA and END-OF-ACTIVITY.

data - specifies the Ph-interface-data (PhID) component of the PhIDU. It consists of one octet of Ph-user-data to be transmitted (Ph-DATA request) or which was received successfully (Ph-DATA indication).

status - specifies either success or the locally-detected reason for inferring failure.

The Ph-DATA confirm primitive provides the critical physical timing feedback necessary to inhibit the DLE from starting a second transmission before the first is complete. The final Ph-DATA confirm of a transmission should not be issued until the PhE has completed the current transmission.

<u>Notifications of PhS characteristics</u>

The PhE has the responsibility for notifying the DLE of those characteristics of the PhS which are relevant to DLE operation. This notification is accomplished by the PhE by issuing a single Ph-CHARACTERISTICS indication primitive at each of the PhE's PhSAPs at PhE start up.

<u>Transmission of Ph-user-data</u>

The PhE determines the timing of all transmissions. When a DLE has a DLPDU to transmit, and the DL-protocol gives that DLE the right to transmit, then the DLE shall send the DLPDU, including a concatenated FCS, by making a well-formed sequence of Ph-DATA requests, consisting of a single request specifying

START-OF-ACTIVITY; followed by three to 300 consecutive requests, inclusive, specifying DATA; and concluded by a single request specifying END-OF-DATA-AND-ACTIVITY.

The PhE signals its completion of each Ph-DATA request, and its readiness to accept a new Ph-DATA request, with a Ph-DATA confirm primitive; the status parameter of the Ph-DATA confirm primitive conveys the success or failure of the associated Ph-DATA request. A second Ph-DATA request should not be issued until after the Ph-DATA confirm corresponding to the first request has been received from the PhE.

<u>Reception of Ph-user-data</u>

The PhE reports a received transmission with a well-formed sequence of Ph-DATA indications, which shall consist of either

a) a single indication specifying START-OF-ACTIVITY; followed by consecutive indications specifying DATA; followed by a single indication specifying END-OF-DATA; and concluded by a single indication specifying END-OF-ACTIVITY; or

b) a single indication specifying START-OF-ACTIVITY; followed by consecutive indications specifying DATA; followed by a single indication specifying END-OF-DATA-AND-ACTIVITY; or

c) a single indication specifying START-OF-ACTIVITY; optionally followed by one or more consecutive indications specifying DATA; and concluded by a single indication specifying END-OF-ACTIVITY.

This last sequence is indicative of an incomplete or incorrect reception. Detection of an error in the sequence of received PhPDUs, or in the PhE's reception process, disables

further Ph-DATA indications with a class parameter specifying DATA, END-OF-DATA, or END-OF-DATA-AND-ACTIVITY until after both the end of the current period of activity and the start of a subsequent period of activity have been reported by Ph-DATA indications specifying END-OF-ACTIVITY and START-OF-ACTIVITY, respectively.

In the first two cases, the DLE concatenates the received data and then attempts to parse it into a DLPDU followed by a concatenated FCS. In the last case the DLE simply discards all reported data.

Functions of the DLL

The functions of the DLL are those necessary to bridge the gap between the services available from the PhL and those offered to DLS-users. When used in a normal OSI environment, the necessary functions of the DLL are those specified in ISO/IEC 8886. When used in a Time Critical OSI environment, the necessary functions are a super set of those specified in ISO/IEC 8886; the enhancements are primarily in

a) the availability of a confirm primitive and a service confirmation deadline for each connection-oriented and connection less data-transfer DL-service;

b) the ability to defer and schedule such data-transfer DL-services;

c) the efficient distribution of DLS-user data from a publishing DLS-user to a set of subscribing DLS-users;

d) the provision of a synchronized sense of internal time among

the DLEs and available to the DLS-users of the extended link; and

e) the standardized availability of local DL(SAP)-address buffer and queue management capabilities.

Connection less data transfer functions

The purpose of the connection less data transfer functions is to transport DLSDUs of limited size between one DLS-user and one or more other DLS-users on the same link, without the requirement for establishing or maintaining a DLC with each of those other DLS-users. This purpose is achieved by means of transmission of DLPDUs providing transfer of a limited amount of user data to one or many DLS-users, with minimal protection against loss of the DLSDU, or misordering of successively-transmitted DLPDUs.

Connection-oriented functions

Connection-oriented functions provide for the establishment,use, resynchronization and abrupt termination of a connection between DLS-users on an extended link. The type of the connection may be selected to support user-data flow either

a) bidirectionality between two peer DLS-users, or

b) unidirectionally from one peer DLS-users to another, or

c) unidirectionally from one publishing DLS-user to zero or more subscribing DLS-users.

The features of the connection may be selected to support transfer of DLSDUs of a negotiated maximum size, with either

1) reliable in-sequence non-duplicated delivery with reset on DLSDU loss, or

2) reduced delay, potentially out-of-sequence but non-duplicated delivery, and reset on DLSDU loss, or

3) minimal delay with in-sequence non-duplicated delivery, but with potential DLSDU loss,

4) minimal delay, unsequenced delivery, potential duplication, and no notification of DLSDU loss, or

5) no transfer in one specified direction.

It is to be noted here that in these latter cases (3-5) the connection can be pre-established. This is used most frequently for unidirectional data flow.

Connection establishment phase

The purpose of the connection establishment phase is

a) to establish a DLC between two DLS-users. The establishment of a publishing DLC is best modeled as the concurrent independent pair-wise establishment of the DLC between the common publisher and each separate subscriber.

b) to determine QoS attributes of the DLC, and

c) to distinguish between DLCs.

Data transfer phase

The purpose of the data transfer phase is

a) to transport DLSDUs between two DLS-users connected by a DLC. This purpose is achieved by transmission of DATA/UNITDATA (DU) DLPDUs, which may involve segmenting of DLSDUs for conveyance in multiple DLPDUs and reassembly by the destination DLE.

b) to resynchronize the flow of DLSDUs between the DLS-users, and notify those DLS-users of information loss after an unrecovered error.

Connection termination phase

The purpose of the connection termination phase is to

terminate abruptly a connection between two or more DLS-users and convey the reason for the termination.

Time synchronization function

The purpose of the time synchronization function is to provide a shared moderate-resolution (about 1/32 ms, or 32 kHz) approximately-synchronized internal time reference for all DLS-users, with a period of at least 26 hours, consisting of two components :

a) a component which increases monotonically with time, with a value of zero at the start up of the local end system, and

b) a second component which, when added to the first, causes the sum to be approximately equal to that of the other correctly-functioning DLEs on the extended link.

Functional classes

In the DLL protocol specification of fieldbus Standards, a DLE's functional class determines its capabilities for autonomous DLL activity, and thus the minimum complexity of conforming implementations. Each class includes all lower-numbered classes. The three functional classes, in order of increasing complexity, are

a) Basic

b) Link Master (LM)

c) Bridge

All functional classes support all DLS-user services and are completely interoperable.

Basic class

The Basic class includes the basic protocol elements of procedure necessary

a) to provide interoperability when responding to DLPDUs sent by a DLS-peer or a bridge DLE,

b) to initiate, reset and terminate DLCs with a peer DLE, to support the orderly conveyance of DLSDUs,

c) to send and receive connection less and connection-oriented DLSDUs, and

d) request services from the LAS,

e) to execute a sequence of link operations contiguously, and

f) to optimize local use of the link.

This class is the minimum necessary for field bus interoperability.

<u>Link Master class</u>

The Link Master class also includes the protocol elements of procedure necessary

a) to cooperate with similar DLEs in establishing and sharing master ship of the link,

b) to detect the absence of an LAS on the link and activate the LAS functions within its own node,

and when providing the functions of the LAS,

c) to maintain an ordered access to the shared link communications resource, responding to requests from other DLEs for use of that shared resource, and

d) to serve as the source of internal time for the other DLEs on the link.

This class is necessary for autonomous operation on the link. At least one DLE on the link shall operate in this class.

<u>Bridge (DL-relay) class</u>

The Bridge class also includes the protocol elements of procedure necessary

a) to enable communications between DLEs on a single link which are themselves periodically incapable of communicating directly on the link (that is, fractional duty cycle (FDC) DLEs),

b) to interconnect two or more local links, by bridging them into an extended link, and

c) to provide a common sense of DL- internal time coordinated across the extended link.

This class is necessary when interconnecting two or more local links to form a multi-link extended link, or when one or more DLEs on the local link are fractional duty cycle (FDC) DLEs. When a multi-link extended link exists, the individual local links shall be interconnected by DLEs which operate in this class.

3.4 Fieldbus Application Layer (FAL) [7]

The fieldbus application layer provides the communication capabilities required by time critical distributed applications of real open systems. It is directly supported by the field bus data link layer to transfer FAL PDUs.

The FAL differs from the other layers of OSI in two principal aspects:

* OSI defines a single type of binding mechanism, the

association, to connect APs to each other. The FAL, on the other hand, defines the Application Relationships,of which there are several types, to bind APs together.

* The FAL uses the DLL to transfer its PDUs and not the presentation layer. Therefore, there is no presentation context that can be used by FAL.

Fundamental Concepts

The operation of time critical real open systems is modeled in term of interactions between time critical application processes. The FAL permits these application processes to pass commands and data between them.

Cooperation between APs requires that they share sufficient information to interact and carry out processing activities in compatible manner. These shared information is referred to as the universe of discourse in the terminology of ISO/TR 9007. For application fieldbus the universe of discourse may be restricted to a single fieldbus segment or may span multiple segments. That is, the data used within a fieldbus system may be completely contained within one fieldbus segment, or it may be distributed to more than one segment. This implies that AR end points may be contained solely within one fieldbus segment, or they may be distributed across more than one segment.

The nature of interactions between AP-invocations can be described by four kinds of informations:

* Information that identifies the applications involved in the distributed time critical information processing

activities.

* Information describing the procedure to be used to effect communication between the AP-invocations for the control and coordination of distributed information processing(i.e. protocol).

* Information representing the net effect of past interactions between the AP-invocations(i.e. AP behavior).

* Information representing the time constraints in order to schedule the transfer between APs(i.e. AP behavior).

The FAL standard provides definitions of procedures for interworking which are related to these four kinds of information. The remainder of this section describes the structure of FAL in brief.

Application Processes

Definition of AP

An application process represents the externally visible set of resources, including processing resources within a time critical real open system that may be used to perform a particular information processing activity. Therefore, APs must be configured with communications capability that enable AP to AP interaction.

An AP may organize its interaction with other APs whatever way is necessary to achieve a particular information processing goal.

Application process type

An application process type is a specification the interworking capabilities for a particular class of application processes. All application processes of the same type provide the same set of interworking capabilities.

AP Invocation

A specific utilization of part of or all the capability of a given application process in support of a specific occasion of information processing is referred to as application process invocation. At any particular time an AP may be represented none, one, or more AP invocation. An AP invocation is responsible for coordinating its interactions together with other AP invocations.

Application Object

An application object is an accessible object contained within an application process. Application objects, themselves do not contain communication capabilities and are accessed through those of there AP. AOs may contain other AOs.

Application Entity

Application Entity Type

Application entities provide the communication capabilities for APs. Each AP may be configured with one or more AEs. Each AE provides a set of services and supporting protocols, grouped into application service elements(ASE), to enable communication between APIs.

Application entities that provide the same set of ASE are of same AE type. Two AEs must be of same type in order to communicate

with each other.

Application Relationships Between AEs

When it is involve two or more APIs in execution of a distributed task, a relationship between the APIs for the purpose of communication must exist. Such a relationship is defined as Application Relationship(AR). ARs used for client/service communications relate two AEs, while ARs used for producer/consumer communications may relate more than two AEs.

ARs are modeled as a set of conveyance paths between AEs. Each conveyance path conveys PDUs in the direction between one or more AEs. Each receiving AE on a conveyance path receives all PDUs transmitted by sending AE. The resources of an AE involved in an AR is modeled as a AEI, which has capability of all the services of AE.

Application Service Element

It is a set of application functions that provide a capability for the interworking of application entity invocations for a specific purpose. The application layer in the fieldbus offers several service elements. Some of these are specific to answer user needs while others are used for management.

Management Concepts

An AP may request through the cooperating activity of its own resident AE and other remote AEs, to access the application objects belonging to remote APs for management purposes. This

activity shall result in APDUs being exchanged its AE and other remote AEs.

An AP may also request its own resident AE to access local communication objects for management purposes. This activity may result in APDUs being exchanged between its AEs and other remote AEs. In fieldbus environment objects are abstract elements that represent physical resources, conditional events, and time. These objects are defined according to requirements of the application.

3.5 Logical Devices

General

Devices, communicating in the fieldbus environment provide data and behavior to the outside world. Logical device notion is a tool to provide a specification of the device class of functionalities such as control, maintenance, data sheets...In order to represent its detailed activities the logical device contains APs. The structure of the LDs is set for the purpose of the communication and does not reflect the functional structure of the real device. The actual mapping is of the user layer standard.

Relationship of a LD and Fieldbus Application Layer Structure

According to the application layer structure, a set of external behaviors of device functions is modeled as a LD. A device shall be related to at least one LD if it performs a distributed task in a fieldbus environment. The APs have their own life within the LDs. Some AP within an LD may be active, while other are down. The AP is then the smallest entity the activity of which is atomic. Then when it is intended to perform on line modification in LD, the AP which being modified shall be made un

active from communication stand point. When the modification is finished and tested then the AP can be turned into active.

Relationship of a LD to a "real" Fieldbus device

At least one AP is contained in each fieldbus LD which communicates in the fieldbus environment. Each LD contains a specialized AP dedicated to management function. The scope of this AP manager is the LD.

From a communication stand point LDs can only communicate through the fieldbus, where ever they are located. This feature shows that there is no containment between physical device and the function it performs. Then there is the possibility to move functions across devices without reformatting the communication bindings.

Each AP is related to at least one AE through application object. An AEi represents an instance of communication capabilities of the AE. The binding of an AEi to a set of ARs is described in the referenced paper discussing addressing.

3.6 Communication in a Fieldbus Environment

Overview

The fieldbus is intended to be used in factories and process plants to interconnect primary automation devices (sensors, actuators, field mounted controllers, etc) and to connect these devices with the control and monitoring equipment located in control rooms. This use positions the fieldbus at the lowest

levels of the CIM(computer Integrated Manufacturing) architecture.

Primary automation devices are associated with the lowest levels of the industrial automation hierarchy and perform a limited set of functions within a definite time window. These primary automation devices, also termed field devices are located close to the process fluids, the fabricated part, the machine, the operator and the environment.

These primary automation devices are growing in capability, evolving toward configuration with several sets of functions, to carry out automatically not only their primary activity but also such functions as diagnostics, calibration, and reconfiguration. As simple and intelligent primary automation devices become available, there will be an evolution toward a distribution (in the field devices) of some activities traditionally carried out completely by a centralized system. Some of these activities include diagnostics, data validation, and handling of multiple inputs and outputs pertaining to a small portion of the system.

Examples of these primary automation devices are sensors, actuators, local display devices, annunciators,small logic controllers or small single loop controllers. Fieldbus is intended to connect a wide variety of primary automation devices,exchanging few items of information within a set of bounded time windows, over a limited distance within a building or over a small area of a plant. Some of the expected benefits in using fieldbus are reduction in wiring,increase in amount of data exchanged, wider distribution of control between the primary automation devices and the control room equipment, and the satisfaction of time critical

constraints.

The Fieldbus is subject to many requirements and constraints. Although some of these requirements and constraints are similar to those that apply to higher level networks, many are unique to the Fieldbus and other real-time networks. This introduces the notion of time critical networks.

Fieldbus Requirements

There are certain requirements that influence the design of the Fieldbus:

* Need to support primary automation devices,
* Need for support of time constraints of process,
* Need to support time critical communications,
* Need for variable sharing, data consistency,
* Need for integration in CIM,
* Need for integrity versus time - Quality of service,
* Need to support management services across the network,
* Need to support fieldbus application messaging,
* Need to support high integrity application,
* Need to support user selection of services,
* Performances and capabilities requirements.

Requirement of Fieldbus Application Messaging

The message traffic on the fieldbus can be divided into two general flows. The first of these flows is the horizontal flow. It consists of all of the messaging traffic on the fieldbus between and among the controllers, sensors, and actuators connected to the

fieldbus network. This type of traffic is typically time critical, and may involve multiple peers. It is a need of the fieldbus to guarantee to provide the various devices with a view of the data that is consistent with the original pieces of information.

The second type of flow, the vertical flow needs to support a much more diverse set of traffic. It consists of the message traffic between devices on the fieldbus network and other devices higher in the CIM hierarchy. This traffic typically has different requirements than the horizontal flow. It may be primarily used in a non time critical context and the exchanges may be point-to-point between the supervisory or monitoring equipment and the target device which is observed or commanded. The traffic is prioritized in order to sort urgent traffic from background one. Maximum delays can be assigned but meeting these delays is mainly statistical.

<u>Need to Support High Integrity Application</u>

The fieldbus shall provide optional mechanism for building systems which have different levels of integrity. The following features shall be integrated:

* Error detection (probability of undetected errors very low).
* Fault confinement, a faulty device(including a power supply attached to the data lines) shall not be able to prevent fieldbus communication between other devices.
* Redundancy of common resources (media,bus scheduler if any etc...)

* Data redundancy(same information at disposal of several connected devices with verification of time consistency to allow dynamic voting of different types such as 2x2,for security or 2x3, for availability).
* Full network redundancy(in this case the 2 networks need to be synchronized).
* Device transparent redundancy where two connected devices provide the same operational functions. This will allow that the change over between the two devices will not impact the communication configuration of the other connected devices in relation with them. This supposes that more than one device can share the same physical address(if addressing include this address in its addressing scheme for data access) or data access is logically independent of the physical location. This features will allow building system with the graded level of integrity required by the application foreseen by the user.

Need to Support User Selection of Services

Applications have different needs in terms of priority, service, integrity, behavior and recovery in case of network congestion. This means that services and quality of services shall be user selectable. So for even driven communication, priorities shall be available at user level in order to sort the messages in the flow of data. It can be done by a priority attribute of a service or the communication may provide several services with attached priority. These mechanisms shall go across segment and interconnecting devices when destination is on an upper level of

network.

On the other hand, in case of unsuccessful transfer, the recovery may have different aspects according to the quality required. As a limited no of cases in practical following is anticipated:

*Periodic (graded severity of timeliness), no verification; the user application requests only reporting non compliance.

*Periodic (graded severity of timeliness), verification, the user requests retransmission of inconsistent data.

*Periodic (graded severity of timeliness), coherence of verification; the user requests that all consumers verify their data just received and exchange the results to detect non coherence between them.

*Verified, not periodic; the user requests to know if information sent has been correctly received and decoded by the intended receiver. Optionally when a fault is detected the communication protocol will decide to retransmit the message.

*Ordered, verified, not periodic; the user request to know if the pieces of information sent to the intended receiver have been received in the sequence it has defined. Optionally the communication protocol will decide to retransmit the missing part providing adequate protection for duplicate transmission.

An another quality of service is the capability of the bus of never getting overloaded by means of flow control. As the application have various requirements, the flow control mechanism shall be made available at the user level. This will enable the

user to divide the available time into slots reserved to a specific category of transfers so that when a slot is overloaded by the random messages the other can still transfer scheduled data without additional delay.

A MICROCONTROLLER BASED REMOTE TRANSMISSION UNIT

Fieldbus, as discussed in the previous chapter, specifies data transmission speeds, standard connectors, transmission media, multidrop, multipoint and point to point connection, etc. All these requirements put a limitation on the cost and type of intelligent unit which can be placed in the field, i.e. on the plant floor. The basic features of a remote field unit in typical industrial network are

(1) Conformance to the physical layer specifications.

(2) Transducer excitation, signal conditioning and analog to Digital Converter

(3) Communication interface for the specified data transmission rates and format.

(4) Counters, timers and some mechanism for Event Detection.

(5) Interrupt feature.

(6) Local memory.

(7) Addressability of the field unit.

(8) Error checking and correction.

Fig. 4.1 gives the functional block diagram of a Remote Transmission Unit. In this figure a signal conditioning and multiplexer unit receives inputs from sensors and actuators. These inputs are transferred to the microcontroller based system after signal conditioning and multiplexing. The microcontroller unit further transfers these data to the Field Communication Unit

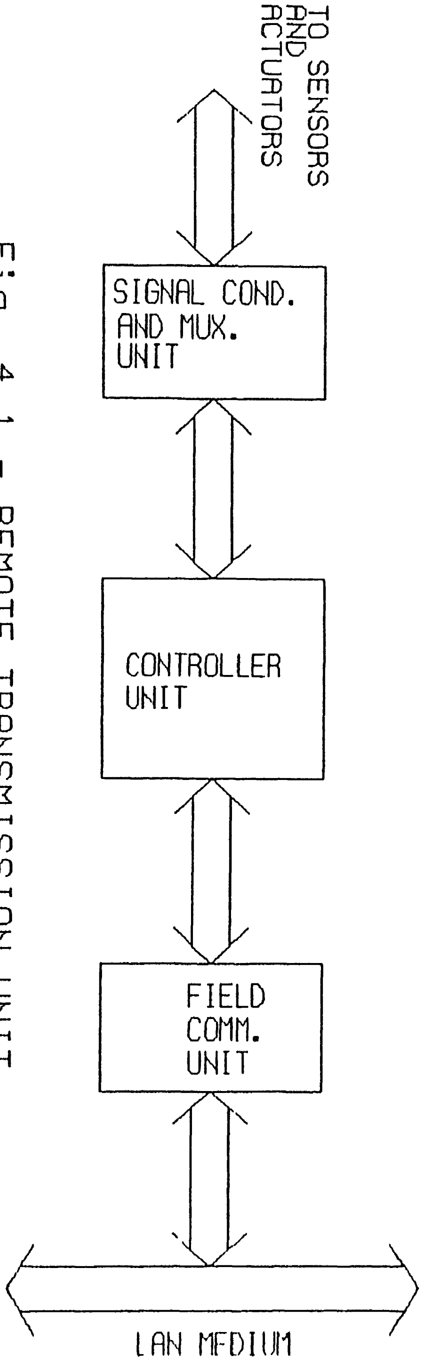


Fig. 4.1 - REMOTE TRANSMISSION UNIT

(FCU), which is resident on a LAN. Functions of FCU may vary depending on the type of LAN used.

In the present work, an experimental unit based on MC68HC11E9 microcontroller has been designed. This unit can monitor a process and support a two way digital communication link. It can be used for control applications with a few modifications.

In the present chapter a block diagram of MC68HC11E9 microcontroller based unit is presented, Fig. 4.2, and functions of blocks of this diagrams are discussed. Section 4.4 gives chart of the software and a brief explanation of these flow charts.

4.1 Microcontroller

The designed remote data acquisition and transmission unit uses, high-density complementary metal-oxide semiconductor MC68HC11E9 microcontroller, which is an 8 bit MCU with highly sophisticated, on-chip peripheral capabilities. This can be changed through software. The HCMOS technology used on the MC68HC11E9 combines the smaller size and higher speeds with low power and high noise immunity of CMOS. Fig 4.3 shows the block diagram of MC68HC11E9 MCU. In this diagram the major sub-systems and how they relate to the pins of MCU is shown. In the lower right-hand corner of this figure the parallel I/O sub-system is shown in the dashed box. The functions of this sub-system are lost when MCU operates in expanded modes, however they can be regained with an additional chip MC68HC24 the port replacement unit.

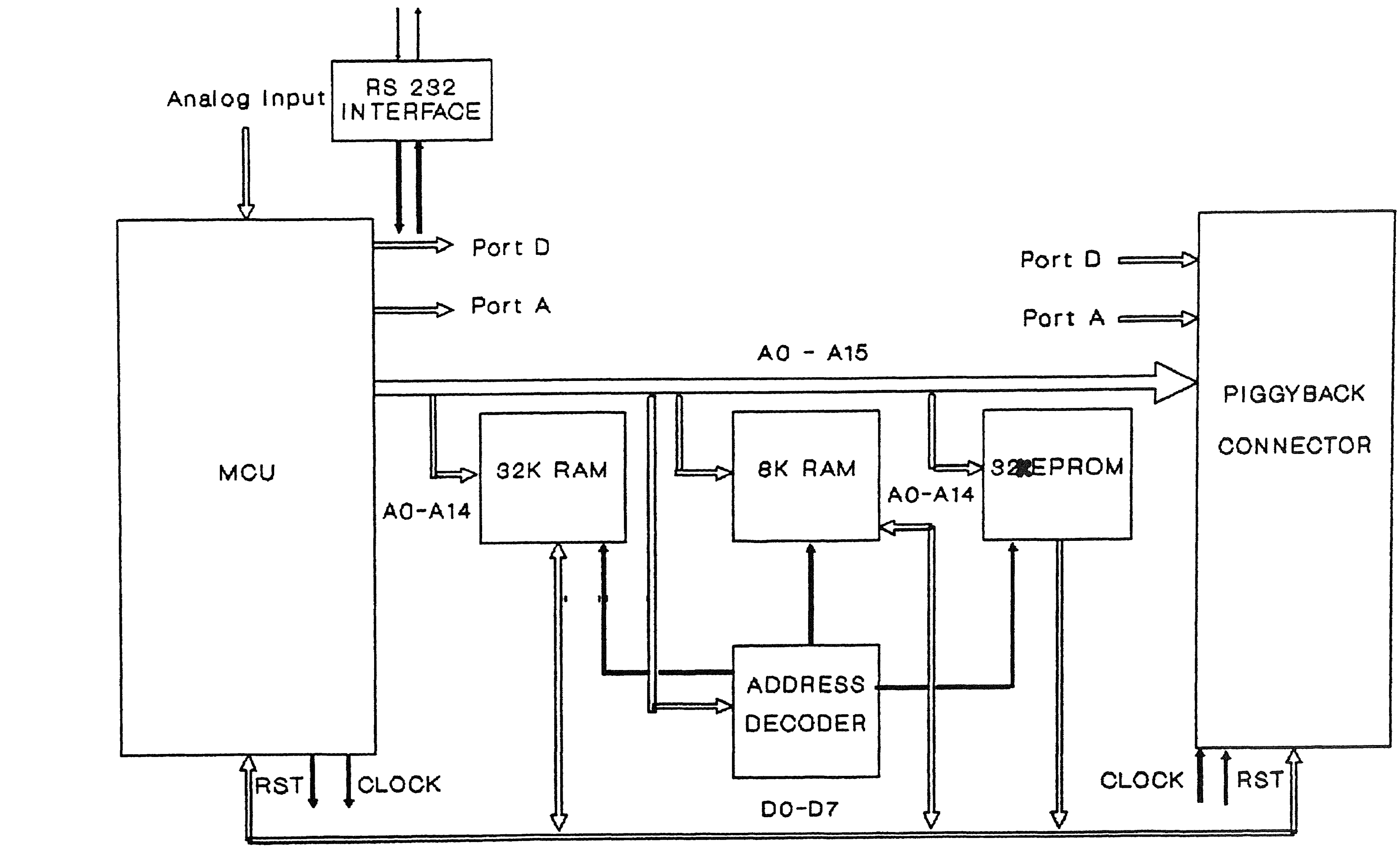


Fig. 4.2. Block Diagram of MC68HC11E9 Based RTU

4.1.1 Configuration and Modes of operations

Hardware Mode Selection : There are two fundamental modes of operation of MC68HC11E9 MCU :

1. single chip mode 2. expanded mode

Each of these modes has a normal variation and special variation. These four modes are selected by the levels on the MOD A and MOD B pins during reset. The special variation of single-chip mode is called special bootstrap mode and the special variation of expanded mode is called special test-mode. The special boot-strap mode allows programs to be downloaded through the on-chip SCI interface into internal RAM for execution. This boot loaded program is used for variety of tasks such as loading calibration values into internal EEPROM or performing diagnostics on the finished module. The special test mode, which is intended primarily for factory testing is seldom used by the user. In the present design this microcontroller is used in expanded normal mode. Table 4.1 gives the summary of the hardware mode selection :

Table 4.1-Hardware Mode Selection of the Microcontroller

Inputs		Mode description	Control bit in HPRIO			
MOD B	MOD A		RBOOT	SMOD	MDA	IRV
1	0	Normal single chip	0	0	0	0
1	1	Normal expanded	0	0	1	0
0	0	Special boot strap	1	1	0	1
0	1	Special test	0	1	1	1

For control bits in HRIO register see [10].

4.1.2 EEPROM-based config. register

The EEPROM based config. register allows additional flexibility to the MCU that would be provided by more complex hardware mode selection structure. This register is used to enable or disable ROM, EEPROM, the COP watchdog system and optionally the EEPROM security feature of the MCU. In the present use of the MCU the register is written such that 512 bytes of internal EEPROM is enabled while internal ROM, EEPROM security and COP watchdog system are disabled. A detailed description of different bits of this register is given in [10].

4.1.3 ON - chip memory

The MC68HC11E9 MCU has

1. 512 bytes of internal RAM
2. 512 bytes of EEPROM and
3. 12 kbytes of factory programmable ROM.

All these memories accept the RAM can be enabled or disabled through the CONFIG register. While the RAM can be relocated in the 64 K MAP of MCU, anywhere on the top of 4K page through INIT register [11], the other memories have fixed locations the memory

map presented in Fig 4.4. The default location of this RAM is from 0000. In the designed card the internal RAM is relocated at A000. The EEPROM in the MC68HC11E9 is fixed from B600 - B7FF. Apart from these memories the MCU unit has 64 bytes reserved, relocatable in any 4K page, register space for controlling its operation. Some of these registers are read only, some are write only and some have time protected write feature. [10] [11]

4.1.4 Central Processing Unit

The MC68HC11E9 CPU can execute all M6800 and M6801 instructions (source and object code compatible) and more than 90 new instructions opcodes. The architecture of MC68HC11E9 causes all peripheral, I/O and memory locations to be treated identically as locations in 64 K memory MAP. Thus there are no special instructions for I/O that are separate from those used for memory.

<u>PROGRAMMER'S</u> <u>MODEL</u> :

7 ACC. A 0 7 ACC. B 0	A:B
15 DOUBLE ACCUMULATOR D 0	D
15 INDEX REGISTER X 0	IX
15 INDEX REGISTER Y 0	IY
15 STACK POINTER 0	SP
15 PROGRAM COUNTER 0	PC

MODES : There are six addressing modes used to refer the memory (a) Immediate (b) direct (c) extended (d) indexed (e) inherent (f) relative.

In the indexed addressing mode, use of Y register always adds to one additional byte in the opcode as compared to the use of X register.

4.1.5 I/O Ports of MCU

The MCU MC68HC11E9 has five ports and 40 pins out of 52 pins are dedicated to these ports. These ports are designated as A, B, C, D and E. Each port has its own data direction register, which should be written before the use of the port. Some of these ports are input only, some output only and some port bits can be configured independently for different operations. In the present design port B and C are not available for I/O operation as the chip is used in the expanded mode. Port D is used for communication purpose while port E is used as an analog input port for A/D converter. Appendix B gives the timing diagrams of these ports and a detailed description with the pin logics is given in [10].

4.1.6 Communication Through MCU

The MC68HC11E9 includes two independent serial sub-systems for communications (a) The serial peripheral interface (SPI) [10] is used to allow the microcontroller unit to communicate with the peripheral devices. The SPI is capable of interprocessor communication system in master or slave mode. Data rates as high as 1 Mbits/s are accommodated when system is configured as master

while 2 Mbits/s data rate is achievable when system is configured as a slave.

(b) The serial communication interface (SCI) system : The SCI is a full-duplex UART type asynchronous system using standard non-return-to-zero (NRZ) format (one start bit, eight or nine data bits, and a stop bit). An on chip baud rate generator derives standard baud rate frequencies from MCU oscillator. Both the transmitter and receiver are double buffered; thus, back-to-back characters can be handled easily even if the central processing unit CPU is delayed in responding to the completion of individual character. The SCI transmitter and receiver are functionally independent but use the same data format and baud rate. It is required to provide the external level shifters for RS 232 levels, as shown in the Fig 4.2.

This SCI receiver includes a number of advance features to assure high-reliability data reception and to assist development of efficient communication networks. The MC68HC11E9 resynchronizes the receiver bit clock on all one-to-zero transitions in the bit stream rather than just at the beginning of the start bit time; therefore differences in baud rates between the sending device are less likely to cause reception errors. Three logic-level samples are taken near the middle of each bit time, and majority logic decides the sense of the bit. The receiver also has the ability to enter a temporary standby mode (called receiver wake up) to ignore messages intended for a other receivers. Logic automatically wakes up the receiver in time to see the first character of the next

message. Clubbed with this receiver wake up feature several MCUs can use their SCI subsystems to form a serial communication network.

The SCI transmitter can produce queued characters of idle and break. In addition to transmit data register empty flag, this SCI also provides a transmit complete flag(TC), this can be used to connect the SCI subsystem with a modem.

In the present design SCI is used to connect the card with a PC. Appendix A gives details of different registers used to control SCI operations.

4.1.7 Analog to Digital Converter System

The MC68HC11E9 has aninternal A/D system. It has 8 channel, 8 bit successive approximation converter with ± 1/2 least significant bit accuracy over the operating temperature range. As this A/D converter uses the charge-redistribution technique, no external sample and hold circuits are required. Eight port E pins are used to supply analog input to this converter, each conversion requiring 32 clock cycle, i.e., in the present design each channel requires 16 μsec for conversion. Appendix A gives details of different registers which govern A/D conversion and Appendix B timing diagram. As shown in the lock diagram of MC68HC11E9 MCU, one can change the reference levels of the converter using V_{REFL} and V_{REFH} the minimum and maximum being 0 and 5 volts respectively.

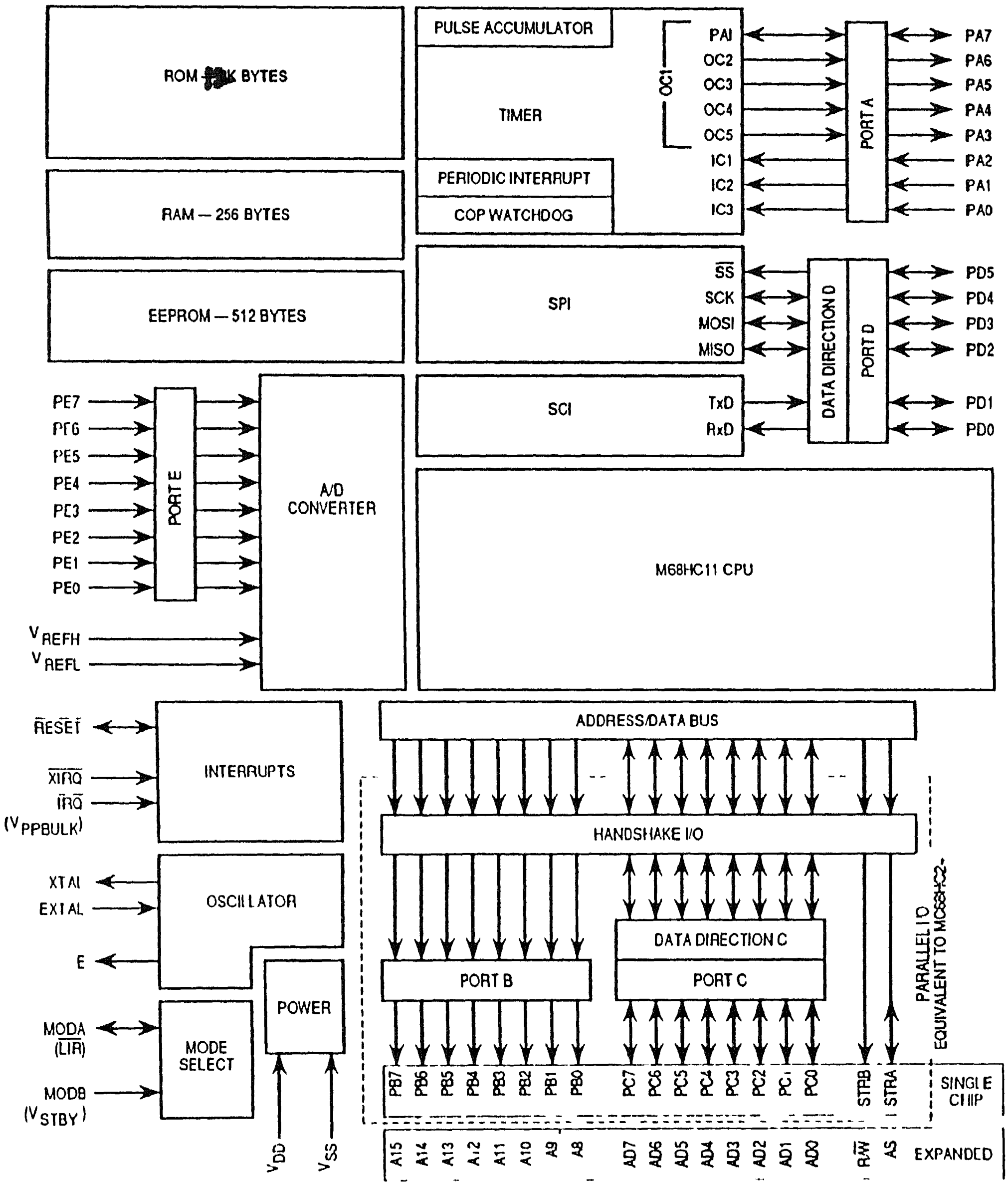


Fig. 4.3 Block Diagram of MC68HC11E9

4.2 Memories and Memory Map

In the designed microcontroller basd unit, the MCU has been used in the expanded mode, hence external memory chips are used. Both the RAM chips used are static RAMs. The table 4.2gives the chip numbers and location of the memory in the map, while fig 4.4 gives the memory map.

Table 4.2 - Memory Chips used in the RTU

Chip Number	Capacity	Address Spac
6206	32k RAM	0000-7FFF
6264	8k RAM	8000-9FFF
27C256	32k EPROM	B900-FFFF

An address decoder block is shown in Figure 4.2, this generates various chip selects for these memories, for internal memories as well as for the Piggyback connector. A combination of programmable logic device and hardware decoder circuitry has been used. Appendix D gives the details of this address decoding circuit.

4.3 Piggyback Connector

A 64 pin piggyback connector is provided in the designed unit to give enough scope for future modifications. This connector has full data bus, address bus(A0-A12), control bus, port A, port D,

reset, clock, interrupt, and power supply connections on its pins. Four partially decoded chip selects are also given on this connector, these chip-selects are used to address 256 bytes in the memory map from B800-B8FF in a group of 64 bytes each. Appendix C gives pin diagram of this connector.

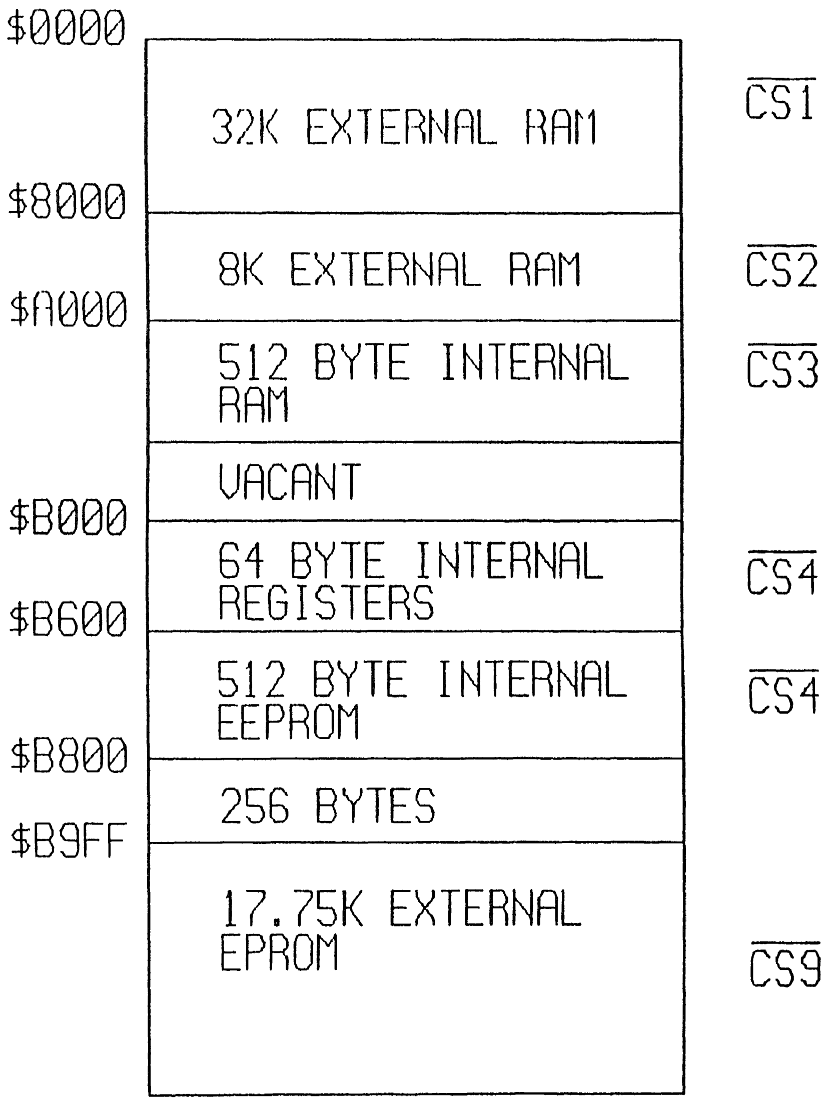


Fig. 4.4 MEMORY MAP

4.4 Description of Flow Charts

This section gives a brief explanation of Fig. 4.6 - 4.9.

Fig. 4.6 gives the flow chart of the main routine written in the assembly language of MC68HC11E9 chip. This routine does the initialization of the microcontroller from power on reset. As mentioned in the section 4.1 , this microcontroller has a few time protected registers and these registers must be written within first 64 cycles of initialization, after power on reset.

Fig. 4.7 gives a flow diagram for the program written in the PC, which communicates with the microcontroller through a serial link. Whenever the microcontroller is required to send data, the PC sends a query byte 'AA'. On receiving this byte the microcontroller SCI system interrupt request is generated, and the microcontroller transmits the channel number and corresponding data (as shown in Fig. 4.9).

Fig. 4.8 gives the flow diagram of the main subroutine. Here microcontroller is used in continuous A/D conversion mode. The MC68HC11E9, PLCC chip has eight A/D conversion channels but only four latches to store the conversion result. Hence conversion is done in a group of four. After writing the A/D Control register the microcontroller checks conversion complete flag, when this flag is set, the latches have valid conversion results and these results are transferred to the memory. After this second group of channels is converted and stored in the memory. This cycle continues till SCI system interrupts for service.

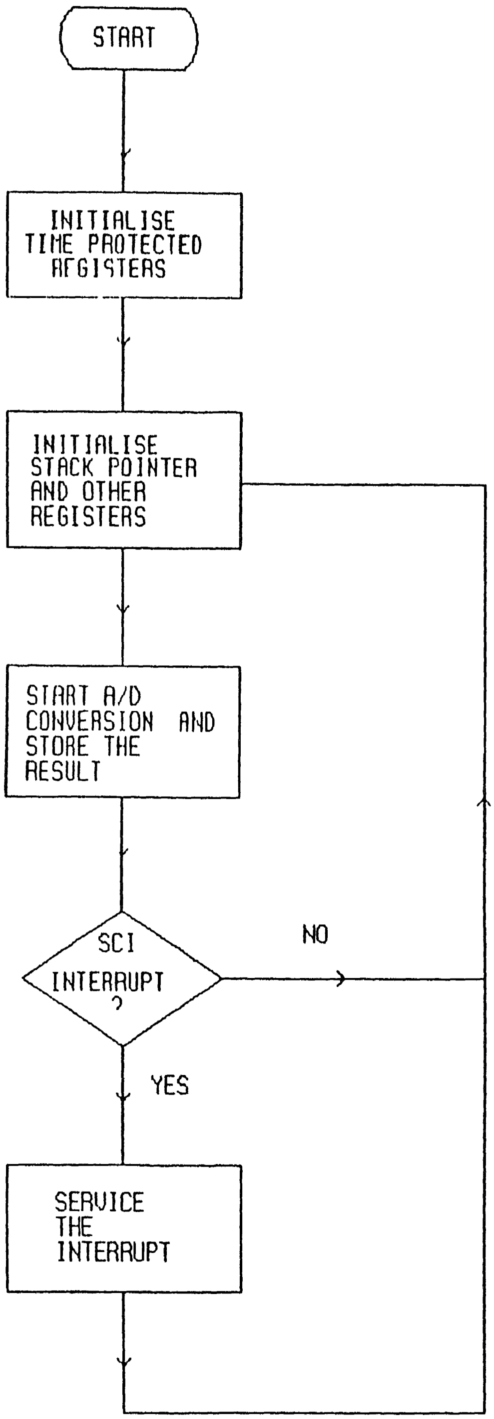


Fig. 4.6 FLOW CHART FOR THE MAIN PROGRAM

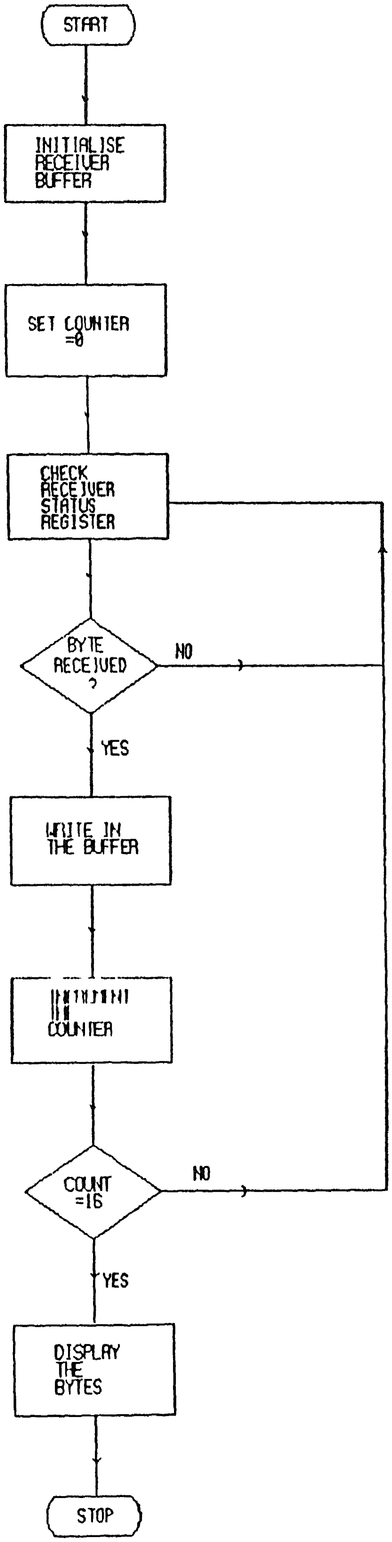


Fig 4 7 FLOW CHART FOR THE RECEIVER COMPUTER

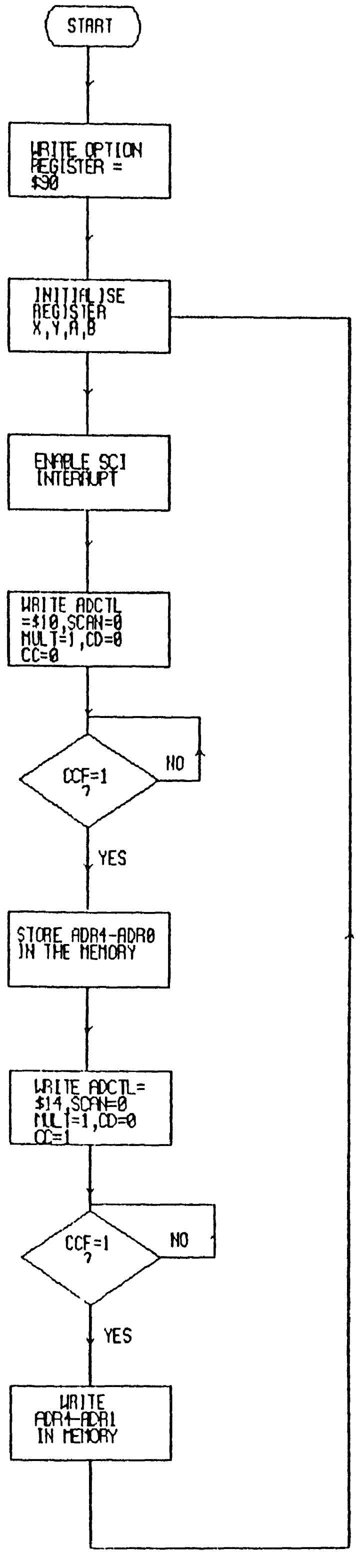


Fig 4 8 FLOW CHART FOR A/D CONVERSION SUBROUTINE

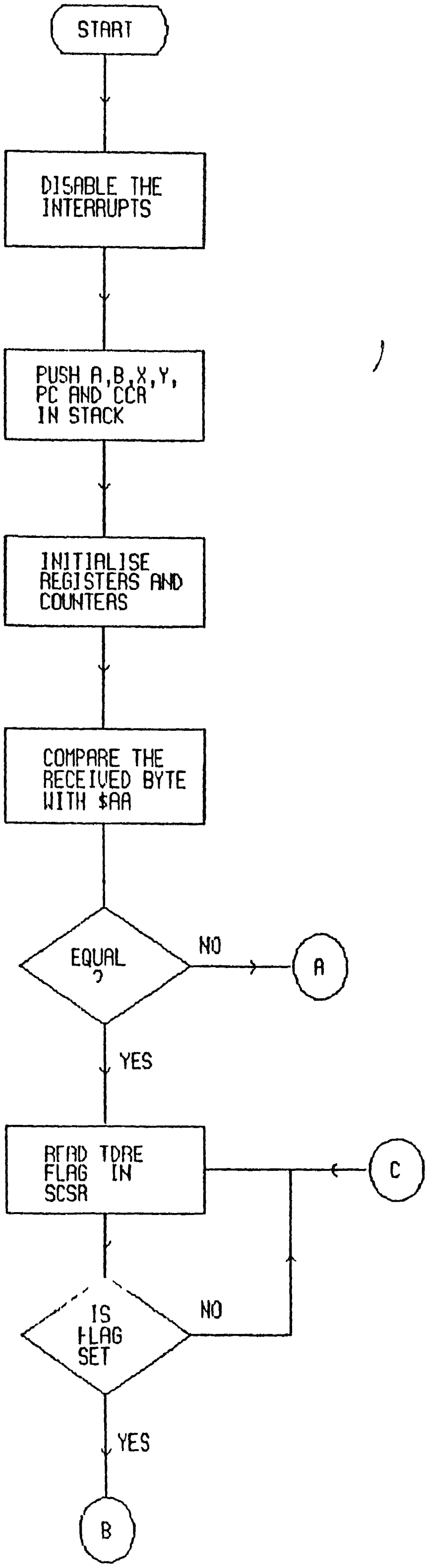
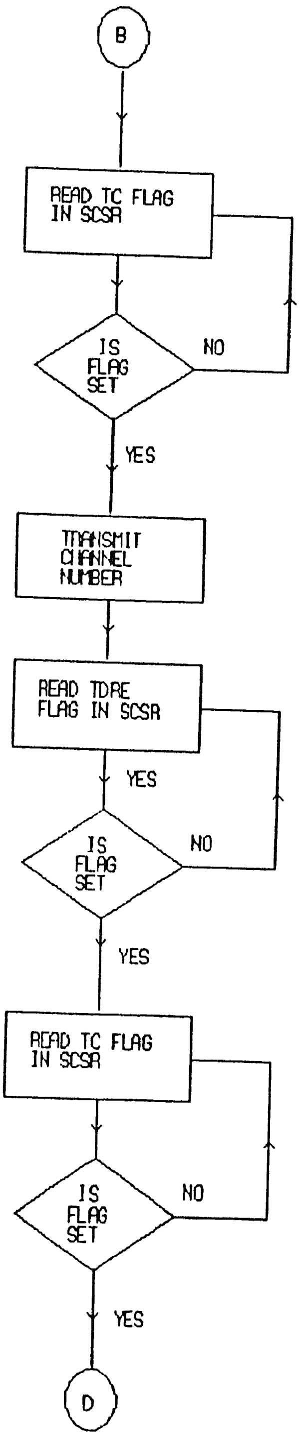
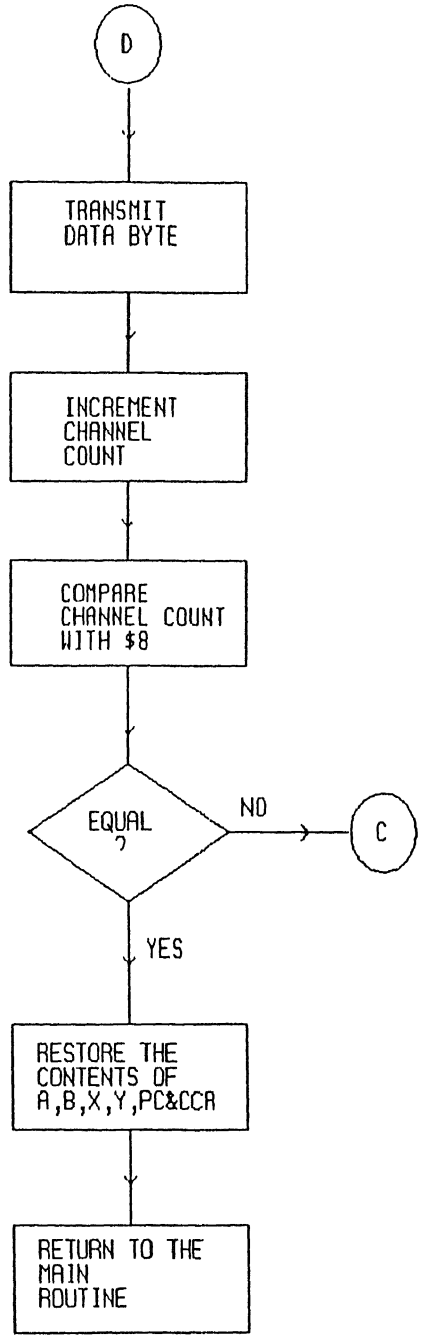


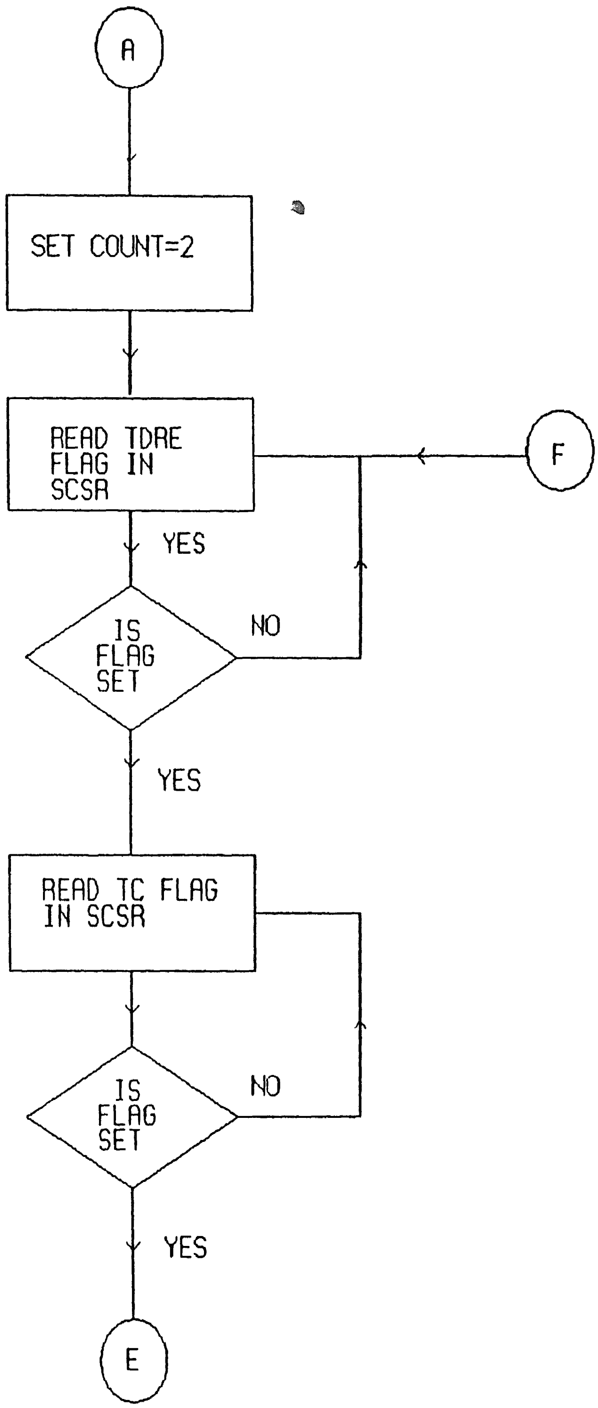
Fig 4 9 FLOW CHART FOR SCI INTERRUPT SERVICE ROUTINE
Sheet 1of5


B
READ TC FLAG IN SCSR
IS FLAG SET
NO
YES
TRANSMIT CHANNEL NUMBER
READ TDRE FLAG IN SCSR
YES
IS FLAG SET
NO
YES
READ TC FLAG IN SCSR
IS FLAG SET
NO
YES
D

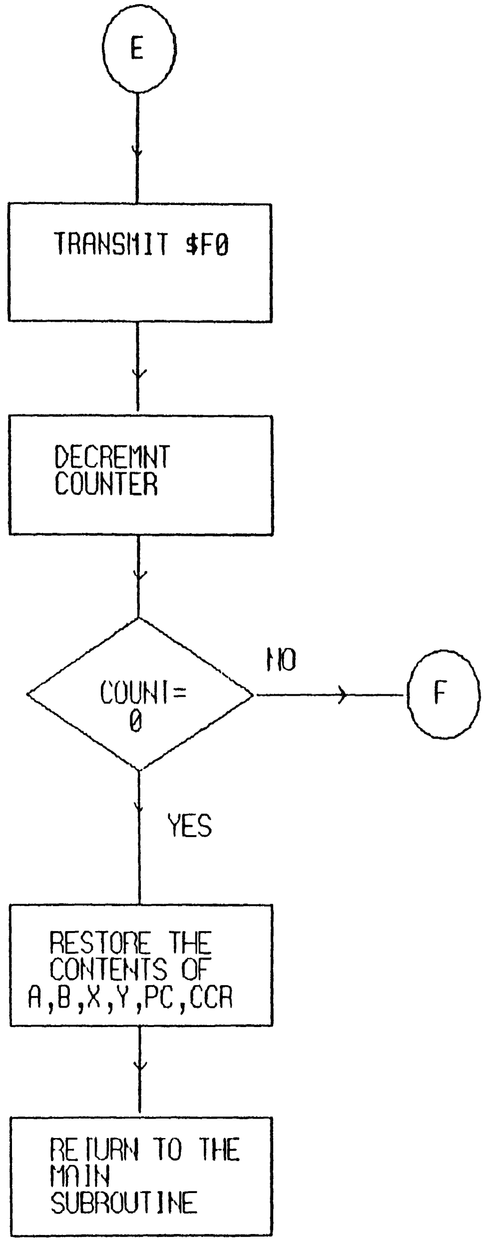
Sheet 2of5


D
TRANSMIT DATA BYTE
INCREMENT CHANNEL COUNT
COMPARE CHANNEL COUNT WITH $8
EQUAL ?
NO
C
YES
RESTORE THE CONTENTS OF A,B,X,Y,PC&CCR
RETURN TO THE MAIN ROUTINE

Sheet 3of5


A
SET COUNT=2
READ TDRE
FLAG IN
SCSR
F
YES
IS
FLAG
SET
NO
YES
READ TC FLAG
IN SCSR
IS
FLAG
SET
NO
YES
E

Sheet 4of5


E
TRANSMIT $F0
DECREMNT
COUNTER
COUNT=
0
NO
F
YES
RESTORE THE
CONTENTS OF
A,B,X,Y,PC,CCR
RETURN TO THE
MAIN
SUBROUTINE

Sheet 5of5

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

The present work can be summarized under the following headings.

Study of fieldbus standards.

A short survey of industrial networks.

Design of an experimental microcontroller based Remote Transmission Unit.

The goal of fieldbus standards is to bring out an open industrial network standard. This is expected to take into account potential future technologies in industrial networks and provide a smooth transition from the plethora of existing proprietary industrial networks towards an open OSI compatible industrial network. Use of fieldbus will help in planning, installation, operation and maintenance.

A short survey of different proprietary industrial networks has been done and various features of these networks have been compared.

An experimental Remote Transmission Unit based on MC68HC11E9 has been designed and tested. This unit converts the analog input, available at portE of the microcontroller, in to digital and transmits it when requested. The piggyback connector and memories on the Remote Transmission Unit can support different future

developments. A synchronous link can be established using the piggyback connector and this can be used to develop a fieldbus test bed.

The MC68HC11E9 chip has five I/O ports, in the present design port B, C, D, and E are used. Port A is free, thiscan be used for Event Detection such as input capture or output comparison purposes. The same port can be used for control applications. It is possible to recover port B and port C, if an additional port replacement unit MC68HC24 is used. This will give more flexibility if the microcontroller unit is used for control applications.

REFERENCES

1. Stallings W., "Handbook of Computer Communications Standards Vol I : Local Networks Standards", Macmillan Publishing Company Newyork, 1987.

2. Uyless Black., "Data Networks : Concepts Theory and Practices", Prentice - Hall, Inc, 1989.

3. Popovic D., Bhatker V.P., "Distributed Computer Control for Industrial automation", Marcel Dekker, Inc, 1990.

4. ISA - S50.02/IEC 1158-2, Fieldbus Part - 2 Physical Layer Specification and Service Definition, 1992.

5. ISA - ds50.02, "Fieldbus Data Link Service Definition", ISA, 1993.

6. ISA - ds50.02, "Fieldbus Data Link Protocol Specification", ISA, 1993.

7. ISA/SP 50 - 1992-457c, "Fieldbus Application Layer Specification", ISA, 1992.

8. Stallings W., "Handbook of Computer Communications Standards Vol 1 : The Open System Interconnection Model and OSI related standards", Macmillan Publishing Company Newyork, 1987.

9. ISO 7489 : 1984, Information Processing Systems- Open System Interconnection - Basic Reference Model.

10. M68HC11 Reference Manual, Motorola Inc., 1990.

11. Advanced Information Manual MC68HC11E9, Motorola Inc., 1988.

APPENDIX A

SCI Control register 1 (SCCR1)

The SCCR1 contains control bits related to the 9 bit data character format and receiver wake up feature. Four of the bits in this register are not used and always read as zeros.

SCCR1 $102C	R8	T8	0	M	WAKE	0	0	0

R8-Receive Data bit 8

When the SCI system is configured for 9 bit characters,this bit acts as the extra (ninth)bit of the RDR.The MSB of received characters is transferred into this bit at the same time the remaining eight bits are transferred from serial receive shifter to the SCDR.

T8-Transmit data bit 8

When the SCI system is configured for 9-bit data characters,this bit acts as the extra(ninth)bit of the TDR.

M-SCI character length

0 = One start bit, eight data bits, one stop bit

1 = one start bit, nine data bits, one stop bit

WAKE- Wake-up method select

0 = Idle line; detection of at least a full character time of idle line causes the receiver to wake-up.

1 = Address mark; a logic one in the MSB position causes the receiver to wake up.

SCI Control Register 2 (SCCR2)

The SCCR2 is the main control register of the SCI subsystem.

SCCR2, $102D

TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK

TIE- Transmit interrupt enable

0 = TDRE interrupts disabled

1 = An SCI interrupt is requested when TDRE is set to one.

TCIE- Transmit complete interrupt enable

0 = TC complete interrupt disabled

1 = An SCI interrupt is requested when TC is set to one.

RIE- Receiver interrupt enable

0 = RDRF and OR interrupts disabled

1 = An SCI interrupt is requested when RDRF or OR is set to one.

ILIE- Idle-Line interrupt enable

0 = IDLE interrupts disabled

1 = SCI interrupt is requested when IDLE is set to one

The idle-function is inhibited while the receiver wake-up function is enabled.

TE- Transmit Enable

0 = Transmitter disabled

1 = SCI transmitter is enabled

RE- Receive Enable

0 = SCI receiver disabled

1 = SCI receiver enabled

While the SCI receiver is disabled, the RDRF, IDLE, OR, NF, and FE status flags cannot become set. If these flags were set, turning off RE does not cause them to be cleared.

RWU- Receiver Wake Up

0 = Normal SCI receiver operation (wake- up feature not enabled).

1 = Places the SCI receiver in a standby mode where receiver related interrupts are inhibited until some hardware condition is met to wake up the sleeping receiver. The condition that wakes the receiver up depends on which method of wake up is specified with the WAKE bit in SCCR1.

SBK- Send Break

0 = Normal transmitter operation.

1 = Enable transmitter to send synchronous break characters.

SCI Status Register (SCSR)

The seven status bits associated with SCI system are located in this register. Some of this bits optionally generate hardware interrupt requests; whereas, others simply indicate errors in the reception of a character. These status bits are automatically set by the corresponding conditions having been met in the SCI logic. Once set, these bits remain set until software completes a clearing sequence.

SCSR
$102E

TDRE	TC	RDRF	IDLE	OR	NF	FE	0

TDRE- Transmit Data Register Empty

0 = Not empty

1 = Indicates that a new character may now be written to SCDR.

TC- Transmit complete

0 = The transmitter is busy sending a character, preamble, or break character.

1 = The transmitter has completed sending and has reached an idle state.

RDRF- Receiver data register full

0 = Not full

1 = A character has been received and has transferred from

the receive shift register to the parallel register where software can read it.

IDLE- Idle line detect

0 = The RxD line is either active now or has never been active since IDLE was last cleared.

1 = The RxD line has become idle.

OR- Overrun error

0 = No overrun error.

1 = Indicates that another character was serially and was ready to be transferred to the register, but the previously received character was not yet read.

NF- Noise flag

0 = No noise detected during reception of the character.

1 = Data recovery logic detected noise during reception of the character.

FE- Framing error

0 = No framing error detected.

1 = A framing error was detected for the character in the received.

A/D Control/Status register (ADCTL)

ADCTL
$1030

CCF	–	SCAN	MULT	CD	CC	CB	CA

CCF- Conversion complete flag

This read only status indicator is set when all four A/D results registers contain valid conversion results. Each time the ADCTL register is written, this bit is automatically cleared, and a new conversion sequence is started immediately. In the continuous scan modes, conversion continues in the round robin fashion, and the result registers are updated with current data even if the CCF bit remains set.

Bit 6- Not implemented, always reads zero.

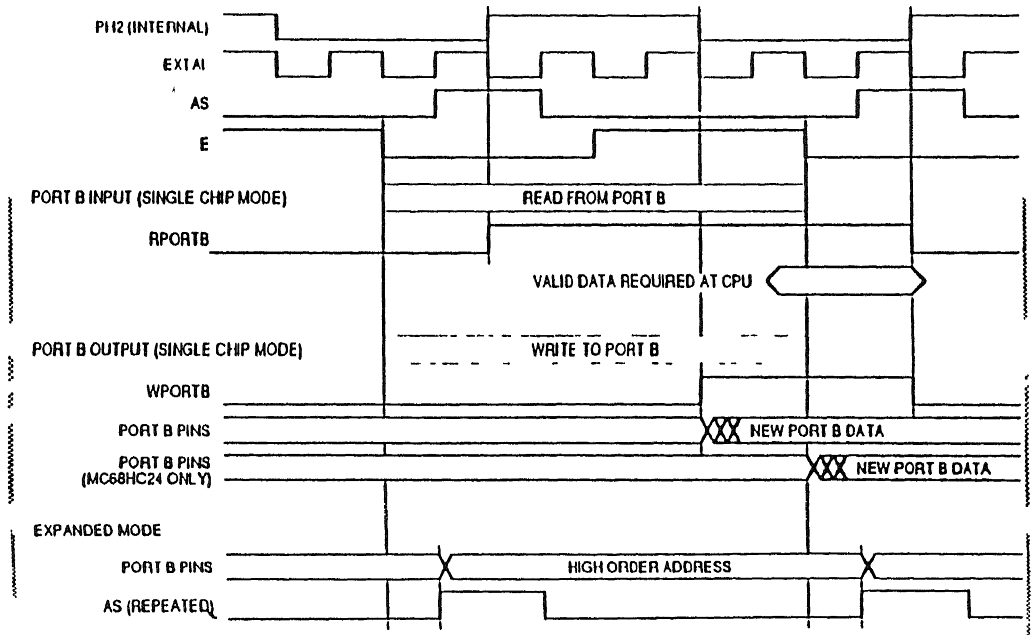
SCAN- Continuous Scan Control

When this bit is zero, the four requested conversions are performed, once each to fill the four result registers. When this bit is one, conversion continue in a round robin fashion with the result registers being updated as new data becomes available.

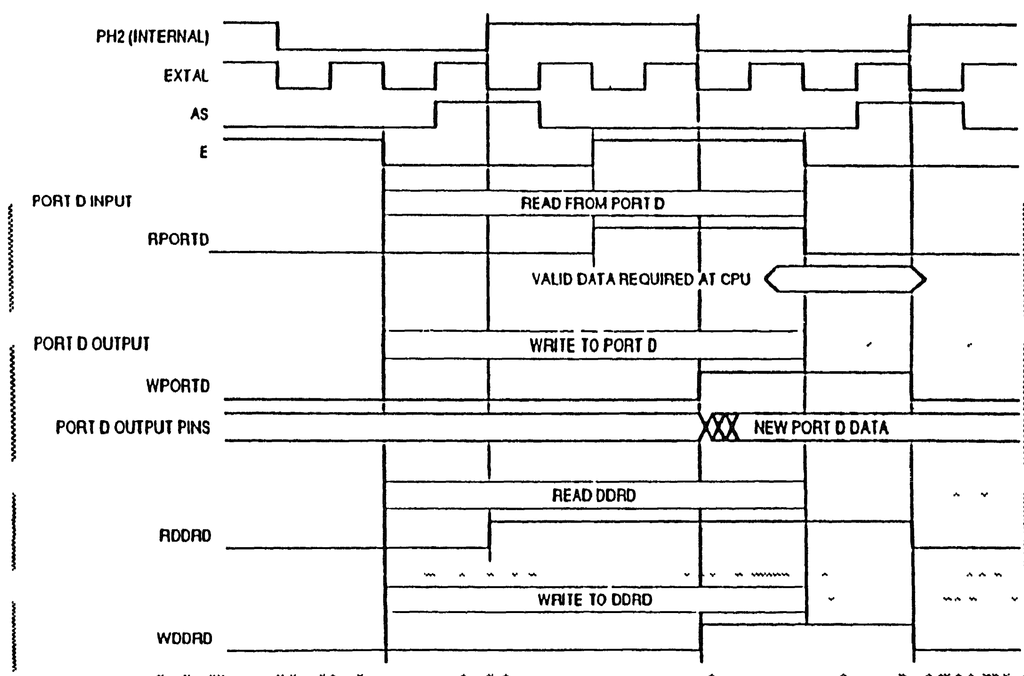
MULT- Multiple channel/Single Channel Control

When this bit is zero, the A/D system is configured to perform four consecutive conversions the single channel specified by the four channel select bits. When this control bit is one, the A/D system is configured to perform conversions on each channel in a group of four channel specified by CD and CC channel select bits.

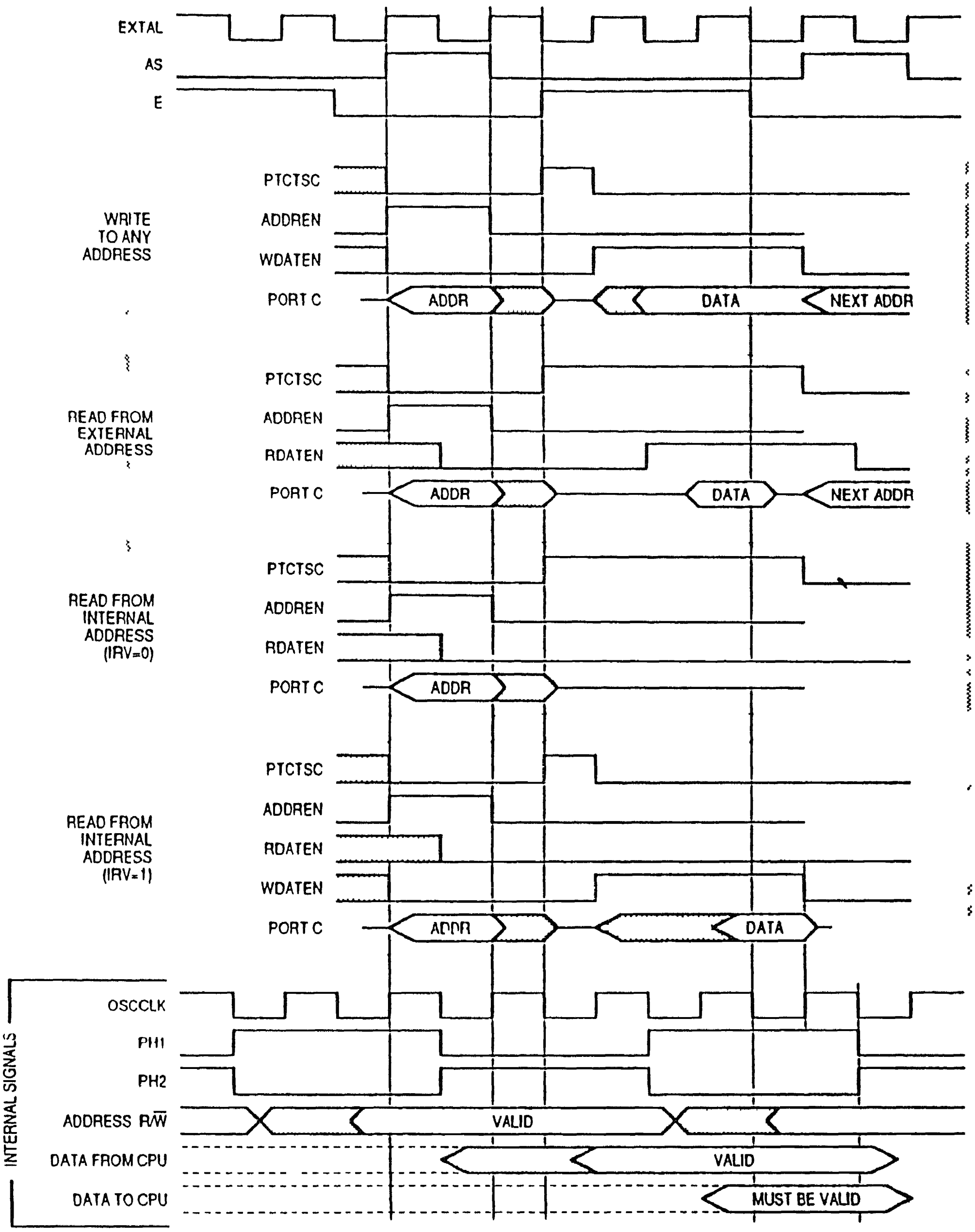
APPENDIX B TIMING DIAGRAMS



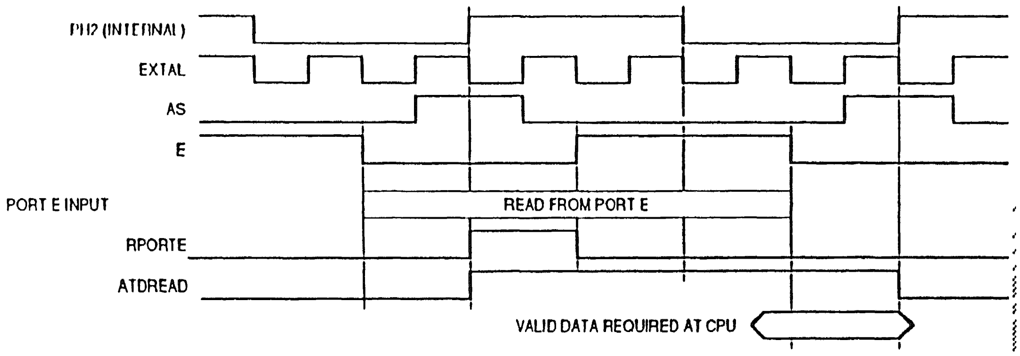
Idealized Port B Timing



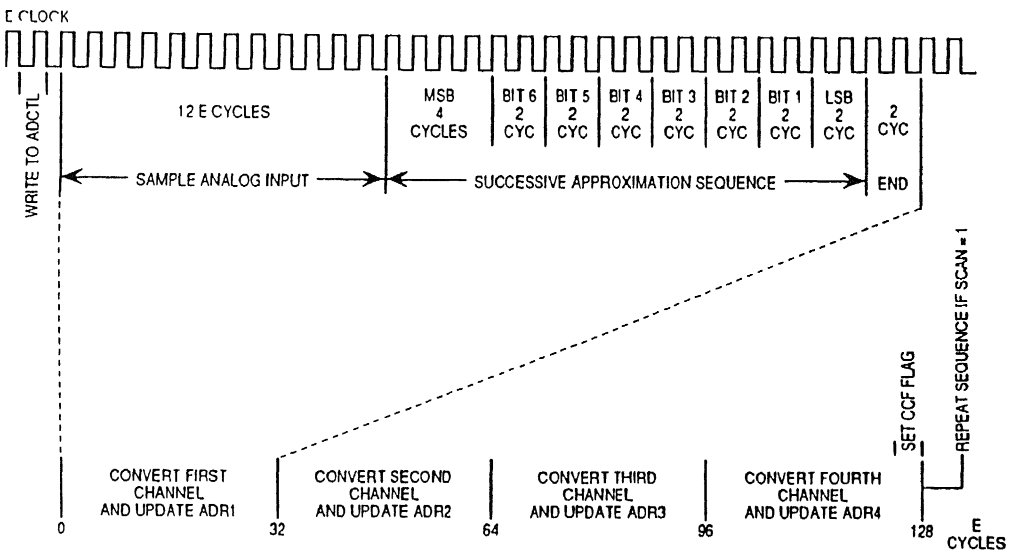
Idealized Port D Timing



Summary of Idealized Port C Expanded-Mode Timing



Idealized Port E Timing



Timing Diagram for a Sequence of Four A/D Conversions

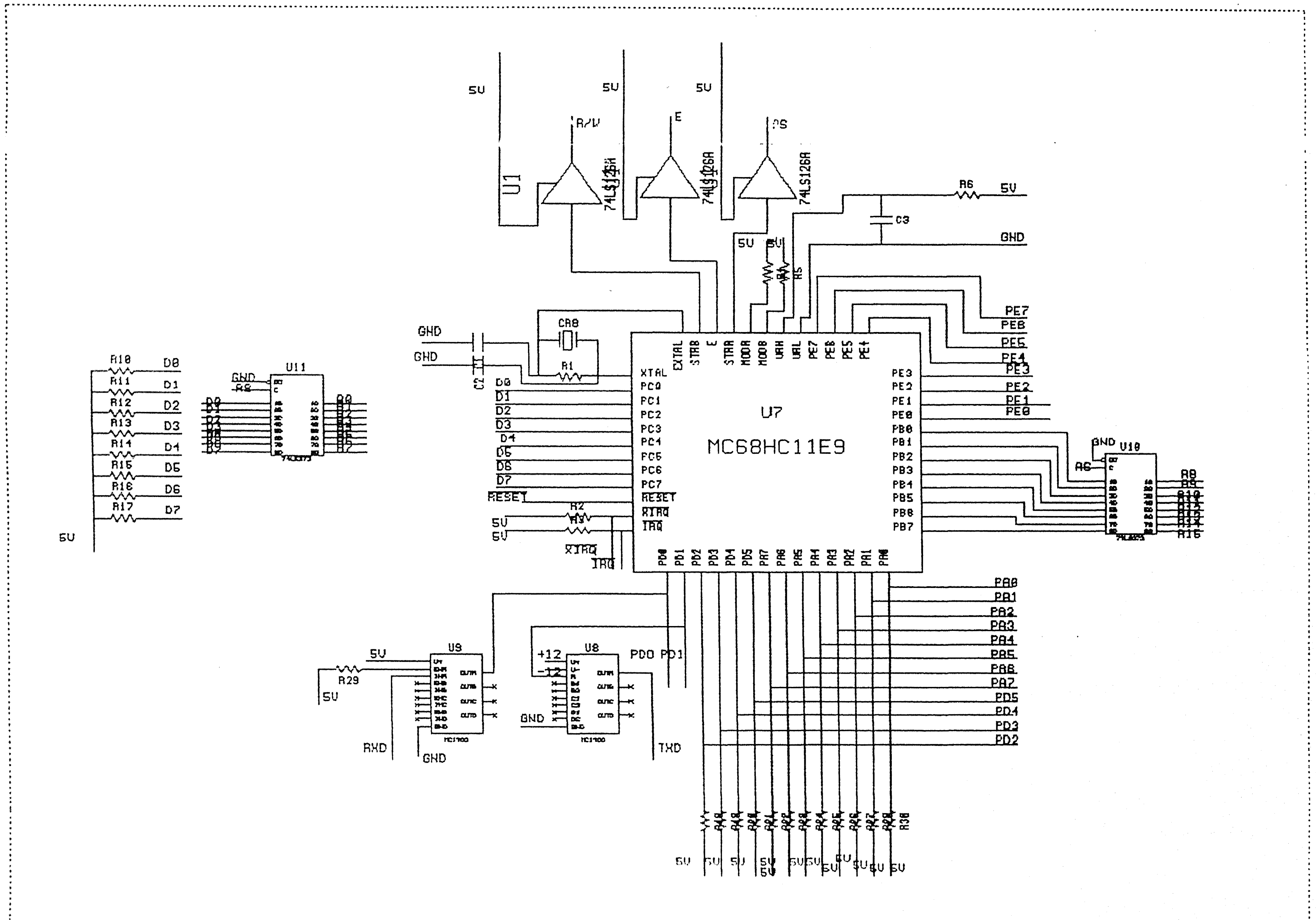
APPENDIX C

PIN CONNECTIONS OF PIGGYBACK CONNECTOR

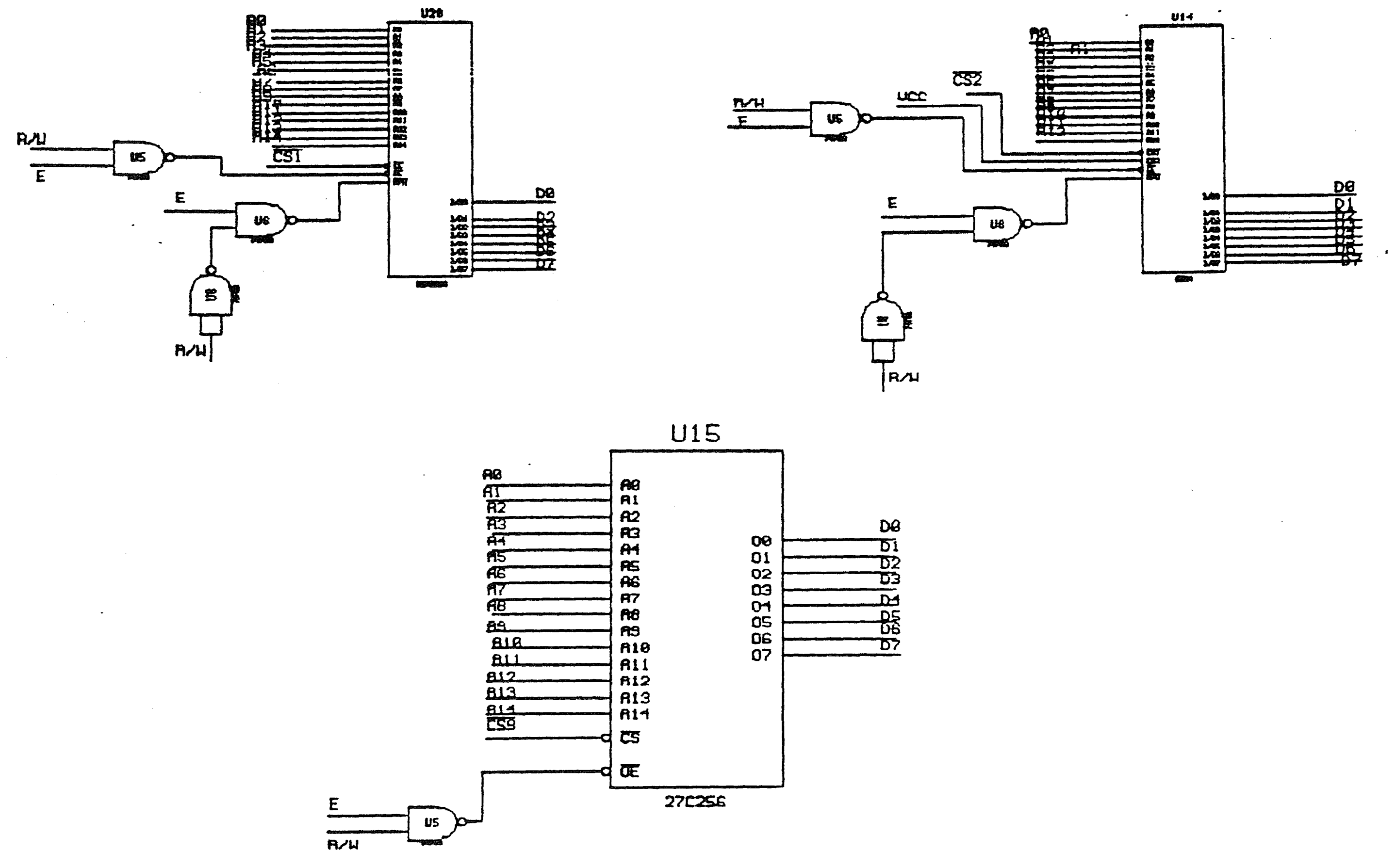
NO	CONNECTION	NO	CONNECTION	NO	CONNECTION
1	GND	23	PA1	45	–
2	5V	24	PA2	46	–
3	D0	25	PA3	47	CS7'
4	D2	26	GND	48	CS8'
5	D4	27	PD2	49	–
6	D6	28	PD3	50	PD1
7	A0	29	E	51	GND
8	A2	30	–12V	52	–
9	A4	31	5V	53	GND
10	A6	32	GND	54	RESET'
11	A8	33	GND	55	PA4
12	A10	34	5V	56	PA5
13	A12	35	D1	57	PA6
14	CS2'	36	D3	58	PA7
15	CS5'	37	D5	59	PD4
16	CS6'	38	D7	60	PD5
17	R/W	39	A1	61	VSTDBY
18	PD0	40	A3	62	+12V
19	AS	41	A5	63	5V
20	XIRQ'	42	A7	64	GND
21	IRQ'	43	A9		
22	PA0	44	A11		

APPENDIX D

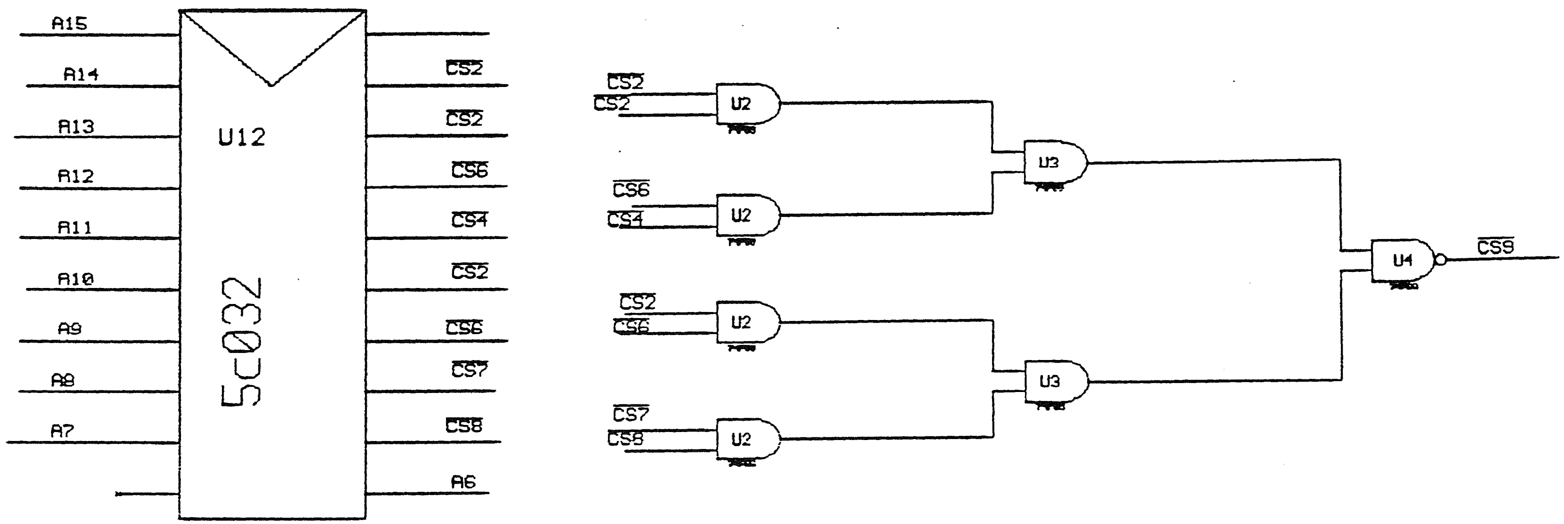
CIRCUIT DIAGRAM OF RTU



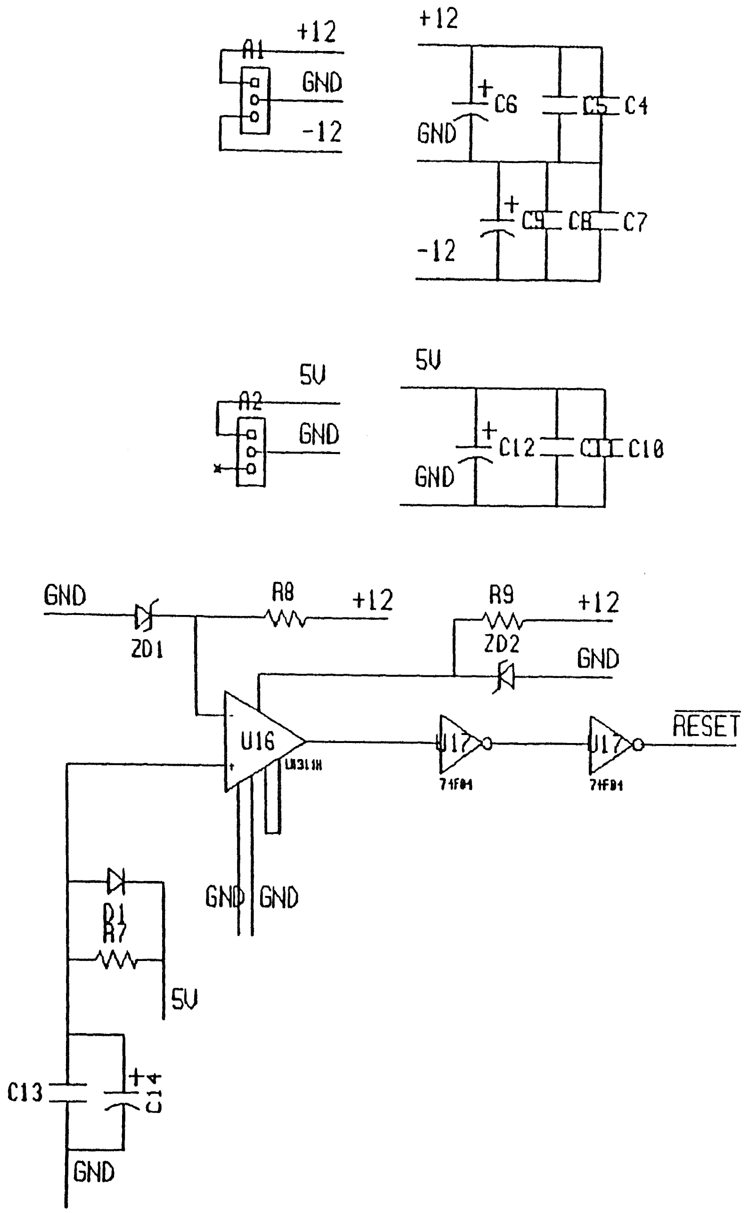
MICROCONTROLLER UNIT SHEET 1OF6



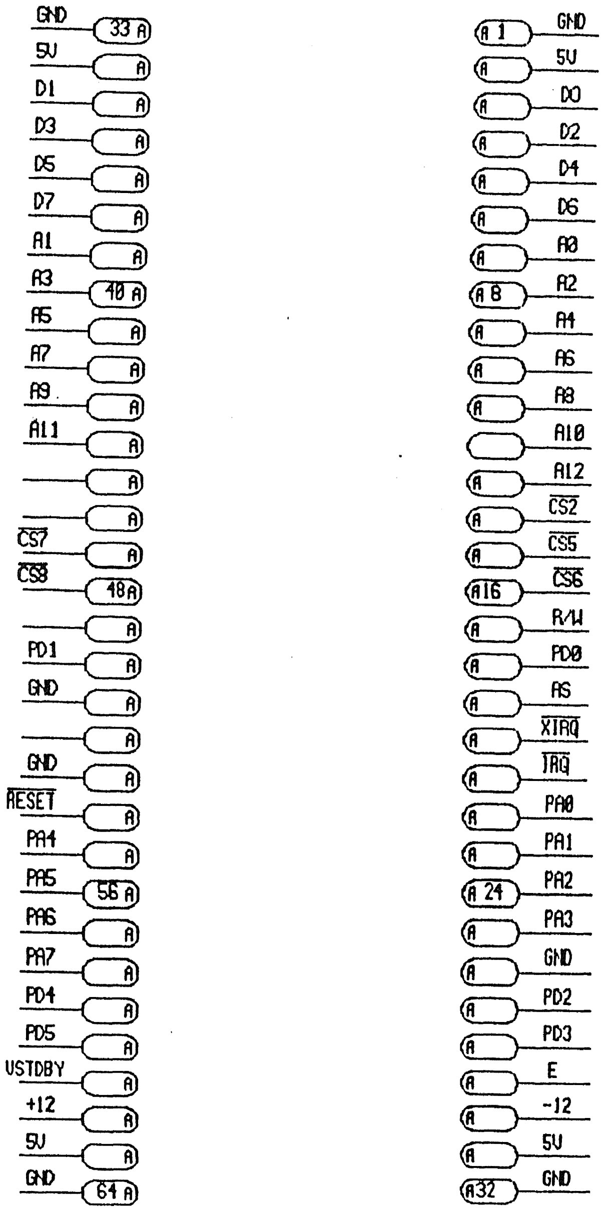
MEMORY CHIPS SHEET 2OF6



DECODER CIRCUITARY SHEET 3ØF6

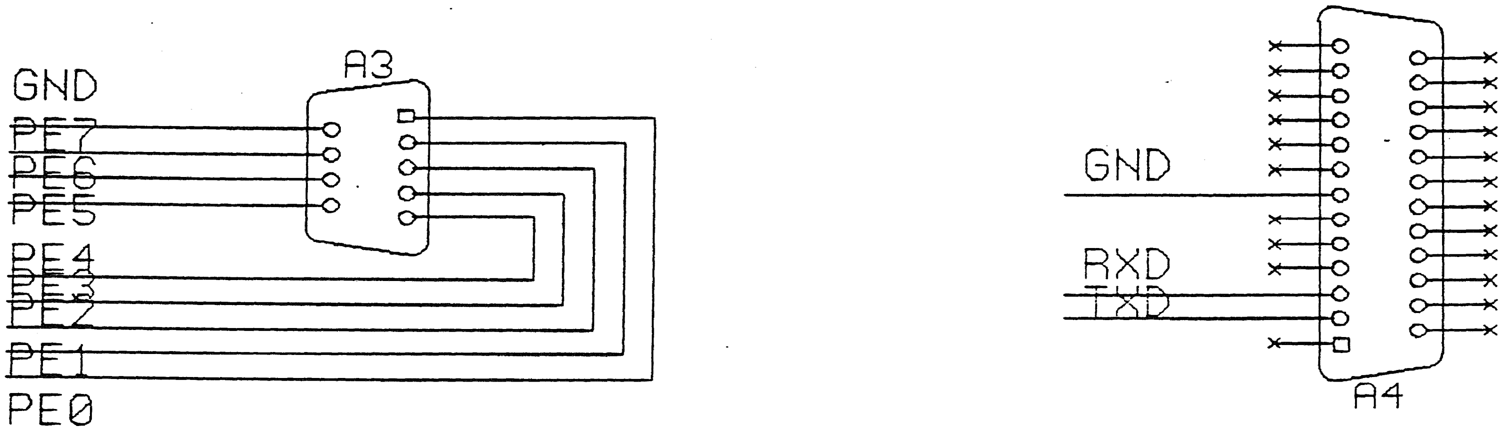


RESET CONTROL CIRCUIT 40F6



PIGGYBACK CONNECTOR

SHEET 5 OF 6



I/O CONNECTORS
SHEET 6OF6